

CARNEGIE MELLON UNIVERSITY

Computational Strategies for the Optimal Operation of Large-Scale Chemical Processes

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree of

DOCTOR OF PHILOSOPHY

in

CHEMICAL ENGINEERING

by

VICTOR M. ZAVALA

Pittsburgh, Pennsylvania

August 28, 2008

Acknowledgments

I'm extremely grateful for all of the opportunities that have arisen and allow me to be writing this dissertation. During my PhD studies, I've been fortunate enough to work with Prof. Lorenz T. Biegler. I thank him for all his patience and guidance. I will always admire his energy, his vast knowledge, and his natural ability to focus on the fundamental problems. I'm in debt to Prof. Ignacio Grossmann and Prof. Erik Ydstie for their support during this time and for being great role models. I thank all the faculty and staff of the Department of Chemical Engineering at Carnegie Mellon for their support under many different circumstances.

I would like to thank Dr. Donald Bartusiak and the process control applications group at ExxonMobil Chemicals for their support during my summer internships. Don has been a great mentor and has helped me to understand the many challenges that arise in industrial practice. I thank Dr. Carl Laird from Texas A&M and Dr. Andreas Wächter from IBM for all their input and help with IPOPT. I thank Prof. Simaan Marwan, Prof. Reha Tütüncü and Prof. Steinar Hauan for their courses. I thank Prof. Thomas Edgar and the faculty of the Department of Chemical Engineering at University of Texas, Austin for their support during the last phase of the PhD. I thank Prof. Antonio Flores-Tlacuahuac and the faculty of the Department of Chemical Engineering at Universidad Iberoamericana, Mexico City for their encouragement to pursue graduate studies.

This work would have never been possible without the unconditional love and support of my mother, my father, and my sister and without the love and patience of my beloved wife Audra. Audi has been strong enough to stand such a terrible husband and to be there even in the most difficult times. I thank my in-laws for their support and I thank all my friends in Mexico and in the US for all the great memories and for teaching me many things.

Abstract

This dissertation deals with the development of computational strategies that enable the implementation of sophisticated first-principles dynamic models in on-line chemical process operations. During the last years, it has been recognized that first-principles models can lead to a tighter integration of the decision-making hierarchy and, consequently, to an increased process performance. The development and implementation of model-based applications involves the extensive use of parameter estimation, state estimation, data reconciliation, model predictive control, and real-time optimization techniques which require the solution of optimization problems constrained by differential and algebraic equations (DAEs). As models of increasing sophistication are considered, the computational complexity of these problems becomes a crucial obstacle. In this work, we present strategies targeted towards this issue. The strategies are based on simultaneous full-discretization approaches and sparsity-exploiting interior-point nonlinear programming (NLP) solvers. In addition, they make use of classical numerical linear algebra techniques and NLP sensitivity concepts.

We first derive strategies for large-scale parameter estimation including standard least-squares and advanced errors-in-variables formulations. Here, we exploit the multi-set structure of these problems inside the NLP solver and derive strategies to extract large-scale covariance information directly from the Karush-Kuhn-Tucker (KKT) matrix. In addition, we establish connections between the numerical properties of the KKT matrix, second order optimality conditions and observability in order to verify the uniqueness of the estimates through the NLP solver.

We then derive on-line synchronization strategies for moving horizon state estimation (MHE) and nonlinear model predictive control (NMPC). These strategies are based on an advanced-step principle which allows to accommodate large-scale dynamic models in on-line environments. Here, we use the dynamic model to *predict* the future state and measurements, use this information to solve reference problems in between sampling times, and *correct* these solutions on-line using NLP sensitivity. This predictor-corrector type strategy allows to minimize the on-line computational time by at least two orders of magnitude and to decouple the MHE and NMPC problems solved in background. We establish rigorous bounds on the loss of optimality, sufficient stability conditions and connections with traditional strategies such as Riccati-like regulators and Kalman filters.

Finally, we implement the proposed computational strategies in the state-of-the-art NLP solver IPOPT and demonstrate the concepts through small-scale case studies. Scale-up and computational performance are demonstrated in a large-scale low-density polyethylene tubular reactor process. In this process, we use a detailed first-principles reactor model to derive an economics-oriented operational framework able to improve its overall profitability.

Contents

Acknowledgments	ii
Abstract	iii
Contents	iv
List of Figures	viii
List of Tables	xii
1 Introduction	1
1.1 Current Practices in Hierarchical Process Operations	1
1.1.1 Planning and Scheduling	2
1.1.2 Target Setting	3
1.1.3 Multivariable and Regulatory Control	4
1.1.4 On-line Estimation and Data Reconciliation	6
1.2 Emerging Practices in Hierarchical Process Operations	7
1.3 Research Problem Statement and Scope of Work	8
1.4 Thesis Outline	9
I Computational Framework	11
2 DAE-Constrained Optimization	12
2.1 Problem Formulation	12
2.2 Approaches to DAE-Constrained Optimization	14
2.2.1 Single-Shooting Approach	15

2.2.2	Multiple-Shooting Approach	16
2.2.3	Simultaneous Full-Discretization Approach	17
2.3	Concluding Remarks	25
3	Interior-Point NLP Solvers	26
3.1	IPOPT Algorithm	26
3.1.1	Computational Issues	28
3.1.2	Inertia Correction	30
3.1.3	IPOPT Software Architecture	34
3.2	Post-Optimal Analysis	35
3.2.1	Extraction of Reduced Hessian Information	35
3.2.2	Nonlinear Programming Sensitivity	39
3.2.3	Fix-Relax Strategy	45
3.3	Concluding Remarks	49
II	Model-Based Operations	50
4	Parameter Estimation	51
4.1	Introduction	51
4.2	Maximum-Likelihood Formulations	52
4.3	Solution with IPOPT	54
4.3.1	Exploitation of Multi-Set Structure	55
4.3.2	Interpretation of Estimates	59
4.4	Concluding Remarks	63
5	Nonlinear Model Predictive Control	65
5.1	Introduction	65
5.2	Standard NMPC Formulation	68
5.2.1	Optimality Conditions	70
5.2.2	Stability Properties	73

5.2.3	Computational Issues	77
5.3	Advanced-Step Formulation	77
5.3.1	Stability Properties	79
5.3.2	Computational Issues	82
5.4	Illustrative Example	83
5.5	Concluding Remarks	88
6	Moving Horizon Estimation	90
6.1	Introduction	90
6.2	Standard MHE Formulation	92
6.2.1	Optimality Conditions	94
6.2.2	Covariance Information	97
6.3	Advanced-Step MHE Formulation	99
6.4	Illustrative Example	102
6.5	Concluding Remarks	104
7	NMPC-MHE Coupling	107
7.1	Standard NMPC-MHE Formulation	107
7.2	Advanced-Step Formulation	110
7.2.1	Stability Issues	112
7.3	Illustrative Example	113
7.4	Concluding Remarks	114
III	Industrial Application	117
8	Industrial Low-Density Polyethylene Case Study	118
8.1	Introduction	118
8.2	Mathematical Modeling and Parameter Estimation	120
8.2.1	Model Structure	121
8.2.2	Model Implementation	124

8.2.3	Model Parameters	126
8.2.4	Case Studies	127
8.3	Nonlinear Model Predictive Control	144
8.3.1	NMPC Problem Formulation	146
8.3.2	Case Studies - Tracking NMPC	148
8.3.3	Case Studies - Economics-Oriented NMPC	152
8.4	Moving Horizon Estimation	156
8.4.1	MHE Problem Formulation	156
8.4.2	Case Studies	159
8.5	NMPC-MHE Coupling	169
8.5.1	Case Studies	169
8.5.2	Computational Issues and Advanced-Step Strategies	170
8.6	Concluding Remarks	179
9	Conclusions	181
9.1	Thesis Summary and Contributions	181
9.2	Recommendations for Future Work	183
A	IPOPT Primal-Dual System	203
B	Riccati Decomposition NMPC	208
C	CSTR Reactor Model	211
D	Riccati Decomposition MHE	212
E	LDPE Tubular Reactor Model	215
E.1	Material Balances	215
E.2	Energy Balances	218
E.3	Model Notation	220

List of Figures

1.1	Example of decision-making hierarchy in chemical process.	2
1.2	Inconsistency of RTO and data-driven MPC controller.	8
1.3	Dynamic Real-Time Optimization (D-RTO).	9
2.1	Schematic representation of multi-stage DAE-constrained optimization problem.	16
3.1	Object-Oriented Implementation of IPOPT.	34
3.2	NLP sensitivity approximation.	41
3.3	NLP sensitivity in the face of active-set changes.	42
4.1	Schematic representation of the relationship between the reduced Hessian and the reliability of the estimated parameters.	61
4.2	Ellipsoidal confidence regions for polynomial parameter estimation example.	63
5.1	Typical NMPC scenario.	68
5.2	Schematic representation of Advanced-Step NMPC controller behavior. . .	82
5.3	Steady-state map between temperature z^t and cooling water flow rate v . Solid vertical lines represent input constraints.	84
5.4	Scenario 1: Effect of plant-model mismatch on the performance of the controllers.	86
5.5	Scenario 2: Effect of noise on the performance of the controllers. Perturbation on residence time at 25% below nominal value.	87
5.6	Scenario 3: Analysis of the effect of mismatch terms on the stability of the advanced step controller.	88
6.1	Schematic representation of estimation horizon for extended MHE problem.	100
6.2	Effect of estimation horizon length N on the states covariance. Concentration z_c is measurement (left). Temperature z_t is measurement (right).	103

6.3	Performance comparison of advanced-step and ideal MHE strategies in the presence of measurement noise.	105
6.4	Performance comparison of advanced-step and ideal MHE strategies in the presence of noise and disturbances.	105
7.1	Scenario arising in NMPC and MHE coupling.	107
7.2	Performance of asMHE/NMPC controller in the absence of measurement noise.	114
7.3	Performance of asMHE/NMPC controller with medium level of measurement noise $\sigma = 5\%$	115
7.4	Performance of asMHE/NMPC controller with large level of measurement noise $\sigma = 7.5\%$	115
8.1	Schematic representation of multi-zone LDPE tubular reactor (top). Typical reactor core and jacket temperature profiles (bottom).	119
8.2	Free-radical copolymerization mechanism of ethylene with a comonomer.	122
8.3	Schematic representation of discretization approach. Reactor core (top) and Jacket (bottom).	126
8.4	Plant and predicted temperature profiles using the base and the proposed estimation strategies for two different grades.	130
8.5	Plant and predicted conversions for estimation approaches analyzed. Results for twenty different grades.	132
8.6	Confidence regions for the parameters of the propagation rate constant k_{p11} . Results with 1 and 3 data sets.	136
8.7	Confidence regions for the parameters of the propagation rate constant k_{p11} . Results with 3 and 6 data sets.	137
8.8	Homopolymer grade macromolecular properties. Plant and model predictions.	139
8.9	Total time and number of iterations for the solution of multi-scenario NLPs with IPOPT. Direct factorization and parallel implementations.	141
8.10	Time per iteration and per factorization of the KKT matrix during the solution of multi-scenario NLPs with IPOPT. Direct factorization and parallel implementations.	143

8.11	Summary of computational results associated with the solution of multi-scenario NLPs with IPOPT. m =number of constraints, DOF =number of degrees of freedom, LB =number of lower bounds, UB =number of upper bounds, It =number of iterations, θ_{total} =total wall clock time, θ_{it} =wall clock time per iteration, θ_{kkt} =wall clock time per factorization of KKT matrix.	145
8.12	Regulatory control structure of LDPE tubular reactors.	146
8.13	Centralized NMPC control structure of LDPE tubular reactors.	147
8.14	NMPC controller behavior. Ramping of heat transfer coefficients (top). Controller response for temperature profile (middle) - gray lines are the responses and dark line is the reference temperature profile. Responses of wall (gray) and jacket (black) temperatures (bottom).	150
8.15	NMPC controller behavior. Closed-loop response of outlet melt-index (top). Response of reactor overall production (bottom).	151
8.16	NMPC controller behavior. Jacket inlet temperatures (top). Initiator flows (middle). Side-streams temperatures (bottom).	153
8.17	Comparison of NMPC and D-RTO controller responses. Closed-loop response of temperature profile for NMPC (top) - dark line is set-point and gray lines are the controller responses. Closed-loop response of temperature profile for D-RTO (middle) - dark line is set-point and gray lines are the controller responses. Reactor overall production responses (bottom).	157
8.18	Relative production gains for individual reactor zones with D-RTO.	157
8.19	Closed-loop response of reactor core temperature profile and position of thermocouples.	159
8.20	Closed-loop response of jacket cooling water and position of thermocouples.	160
8.21	Convergence properties of MHE estimator for perturbations on initial guess. Convergence to heat transfer coefficient of first reaction zones in scenario 1 (top). Convergence to wall temperature profile in scenario 1 (middle) and scenario 2 (bottom).	161
8.22	Convergence properties of MHE estimator for perturbations on initial guess. Convergence to jacket temperature profile in scenario 1 (top). Convergence to radicals concentration profile at time step t_1 (middle). Convergence to polymer melt index at the reactor outlet (bottom).	162
8.23	Effect of estimation horizon length on convergence of estimator. Covariance of wall temperature profiles for different horizon lengths N (top). Decay of maximum eigenvalue of covariance matrix for different measurement scenarios (bottom).	164

8.24	Effect of output noise on performance of estimator. Output noise on reactor core temperature with $\sigma = \pm 3^\circ$ C and convergence to heat transfer coefficients (top). Output noise with $\sigma = \pm 5^\circ$ C and corresponding heat transfer coefficients (bottom).	166
8.25	Effect of <i>unrejected</i> input noise on performance of estimator. True and noisy jacket inlet temperatures for Zone 1 (top). Estimated heat transfer coefficient for Zone 1 (middle). Biased estimates of wall temperature profile (bottom).	167
8.26	Effect of <i>rejected</i> input noise on performance of estimator. True, reconciled and noisy jacket inlet temperatures for Zone 1 (top). Estimated heat transfer coefficient for Zone 1 (middle). Unbiased estimates of wall temperature profile (bottom).	168
8.27	Performance of coupled NMPC-MHE for output feedback. Predicted temperature profile at time step 25 and corresponding profile corrupted with Gaussian noise (top). Convergence of MHE estimator to true wall profile (middle). Jacket water inlet temperature of Zone 2 computed by NMPC controller (bottom).	171
8.28	Performance of coupled NMPC-MHE for output feedback. Convergence of MHE estimator to jacket temperature at a particular location of the reactor (top). Convergence of MHE estimator to wall temperature (middle). Closed-loop responses of NMPC for temperature profile (bottom).	172
8.29	Solution times for NMPC (top) and MHE problems (bottom) with horizons of $N = 10$ time steps.	173
8.30	Scale-up results for solution times of NMPC problem with different horizon lengths.	174
8.31	Comparison of solution times for NMPC problem with tracking and economic objectives.	174
8.32	Comparison of solution times for MHE and EVM-MHE problems.	175
8.33	Effect of NLP sensitivity errors on performance of advanced-step MHE and NMPC strategies in LDPE case study.	178
C.1	Schematic representation of nonlinear CSTR reactor.	211
E.1	Schematic representation of LDPE reactor core-wall-jacket interface.	221

List of Tables

1.1	Comparison of advantages and disadvantages of data-driven and first-principles process models.	6
3.1	Solutions of parametric NLP example.	45
8.1	Summary of available measurements in industrial LDPE reactors.	128
8.2	Optimal objective function values for estimation approaches. Grades A and B cases.	131
8.3	Computational results for single-set NLP problems. On-line parameter estimation case studies. LB=number of lower bounds, UB=number of upper bounds, NZJ=number of nonzeros in Jacobian, NZH=number of nonzeros in Hessian.	134
8.4	Computational results for multi-set NLP problems. Homopolymerization case study. LB=number of lower bounds, UB=number of upper bounds, NZJ=number of nonzeros in Jacobian, NZH=number of nonzeros in Hessian, DOF=degrees of freedom.	138
8.5	Average deviations between plant and model predictions for reactor conversion and grade macromolecular properties for 14 different grades. Homopolymerization case study.	138

Chapter 1

Introduction

In this chapter, we describe the overall context and objectives of the research problem under consideration. In particular, we discuss computational challenges that arise in the incorporation of first-principles dynamic models in the decision-making hierarchy of chemical processes. In addition, we introduce background information and terminology used throughout the dissertation.

1.1 Current Practices in Hierarchical Process Operations

The operation of a chemical process involves a formidable number of decisions that need to be updated in order to compensate for exogenous disturbances (e.g. market prices and product demands) and endogenous disturbances (e.g. unexplainable physico-chemical phenomena) [83, 58]. These disturbances evolve at drastically different time-scales ranging from a few seconds to months and years. Because of this high level of complexity, one would expect that a *centralized* and *systematic* decision-making entity able to take all these factors into account would achieve the maximum process performance. However, this ideal approach is currently infeasible due to practical issues such as computational complexity. As a consequence, the centralized decision-making entity is commonly decomposed into a hierarchy of decision-makers taking care of separate tasks such as planning, scheduling, set-point setting, multivariable control, single-loop regulatory control, and sensors and actuators. This hierarchy is illustrated in Figure 1.1.

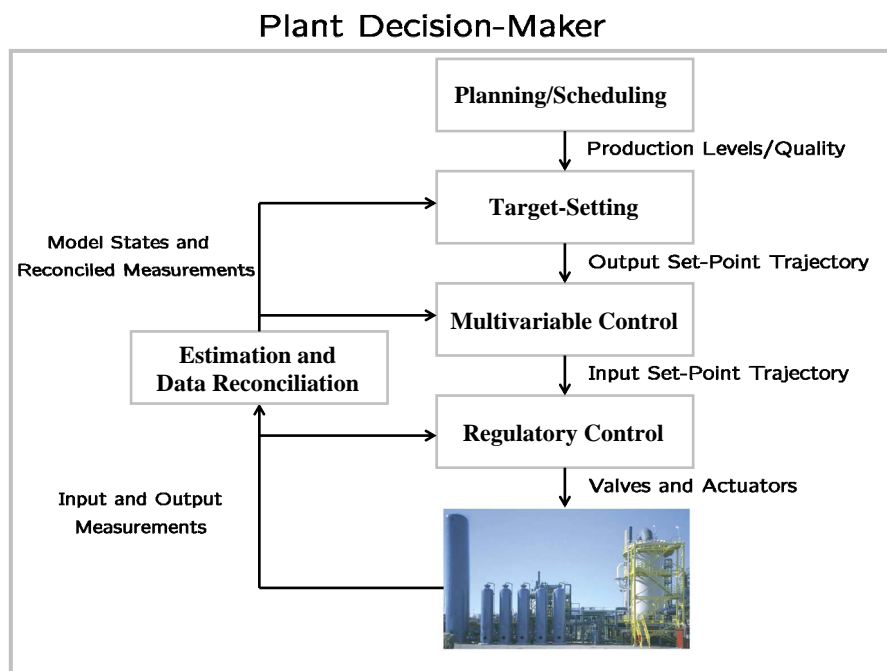


Figure 1.1: Example of decision-making hierarchy in chemical process.

1.1.1 Planning and Scheduling

The planning/scheduling layer makes long-term decisions based on product demands, raw materials, and product prices that change rather slowly (months/weeks). This level uses *forecasts* of exogenous disturbances such as product demands and prices, to find the optimal future policies of the product type(s), production and inventory levels that maximize some measure of the *future* process profitability. Nowadays, the planning/scheduling decisions are obtained systematically from computationally intensive mixed integer programming formulations [55]. To avoid a currently unmanageable complexity, these high-level tasks treat the chemical process as a *black-box* by using rough estimates of its performance limitations (e.g. capacity, transition times). In other words, they do not attempt to manipulate any of the process operating conditions. Consequently, they implicitly assume that a lower-level decision-maker will be able to manipulate the process conditions to achieve the desired

production plan/schedule at all times.

1.1.2 Target Setting

The production plan/schedule is communicated to a target setting layer. Traditionally, this has been a human operator that will adjust the process operating conditions (e.g. output set-points) in order to satisfy the specified production levels and quality. The operating conditions are originally dictated by the nominal design specifications and are refined progressively with experience in order to improve the process performance. At this level, performance is normally measured in terms of *current* process profitability (e.g. product throughput and/or energy consumption). In most cases, experience-based operating policies are *suboptimal* since the process depends on many factors that are difficult to manage simultaneously by the operator.

With advances in computer power, the development of process simulation platforms and advances in nonlinear programming (NLP) algorithms, it has become possible to use first-principles steady-state models to adjust the process operating conditions systematically and in *real-time*. This gave rise to what is known today as real-time optimization (RTO) technology [87]. The main advantage of RTO is that *feasible* operating conditions (e.g. satisfy energy and material balances) can be found through the rigorous model. In other words, RTO allows a *centralized management* of the available raw materials and energy resources. In addition, since the optimization is performed on-line, RTO provides a mechanism to react to changes and *reject* long-term (hours-days) exogenous disturbances. This technology has been taking over the set-point setting layer in many continuous petrochemical processes. The success of RTO technology has been dramatic, leading to millions of dollars in annual savings [122]. However, this technology is only applicable to processes operated under nearly stationary conditions.

Both the RTO and the human operator neglect the inherent process dynamics (e.g. short-

term disturbances). In addition, they do not manipulate any of the dynamic degrees of freedom of the process (e.g. inputs) [117, 118]. Instead, the current output set-point is obtained *assuming* that a lower-level decision-maker will be able to adjust the process inputs to reject any short-term disturbances and keep the process at the desired target at all times.

1.1.3 Multivariable and Regulatory Control

The operating conditions coming from the target setting layer are communicated as set-points of the measured output process variables to some sort of *feedback control layer*. Up to the 1970's, the dominating industrial practice was to communicate each target to a single-loop PID controller. The controller will keep its corresponding output at the desired set-point by rejecting short-term disturbances. A practical problem with single-loop controllers is that they act on a single target without taking into account the multiple interactions arising in a particular unit or in the whole process [101]. As a consequence, these controllers are normally difficult to tune and decouple.

Industrial control practice has shifted almost completely to advanced multivariable control strategies. This has been mostly done in the form of what is known as model predictive control (MPC) [101, 17]. Here, a dynamic process model is used to *forecast* the future impact of the manipulated inputs over the controlled outputs. With these predictive capabilities, it is possible to compute optimal future policies of the manipulated inputs that minimize the transition time to the desired output targets. The main advantage of MPC is that it can handle multivariable interactions implicitly through the model. In addition, it allows to impose constraints directly in the problem formulation which is a convenient way of incorporating process knowledge in the controller. The dominating MPC technology uses input-output, data-driven, linear dynamic models identified from process step responses [101].

A crucial limitation of data-driven dynamic models is that they are constructed around a fixed operating point. In processes operated over a wide range of conditions such as batch processes or continuous polymerization processes, a fixed model is usually not sufficient to achieve an acceptable control performance [123]. The end of the 1980's witnessed the development of first-principles dynamic models for MPC which has given rise to what is known today as Nonlinear Model Predictive Control or NMPC¹. The use of NMPC has brought many important economic benefits, especially in inherently dynamic processes such as batch processes and in continuous processes exhibiting frequent and expensive transitions [10, 123, 48]. As seen in Table 1.1, the use of first-principles models in MPC presents important advantages but also gives rise to important practical issues.

From a computational point of view, data-driven models are manageable since they are described over a relatively small state-space including the measured process outputs. These dynamic models are usually described in discrete-time algebraic form. First-principles models, on the other hand, involve a large state-space including both measured outputs and many unmeasured variables. These models follow fundamental conservation laws which are described by computationally intensive Differential and Algebraic Equations (DAEs). While sophisticated models are currently used in NMPC applications, the computational complexity of these models is still a crucial factor blocking the widespread use of NMPC [15, 10].

Since the MPC or NMPC input actions already take multivariable interactions into account, they can be communicated as separate input set-points to single-loop controllers. Each controller manipulates the valve actuator to obtain the desired input set-point. High-frequency measurement sensors (e.g. temperatures) and low-frequency laboratory measurements (e.g. product quality) are used to monitor the process and close the control and target setting loops.

¹The term *NMPC* alludes to any generic nonlinear model (first-principles or data-driven). Here, we make an explicit distinction between first-principles (NMPC) and data-driven models (MPC).

	Data-Driven Models	First-Principles Models
Development Cost	Low	High
Portability	Low	High
Window of Validity	Narrow	Wide
Control Scope	Measured	Measured and Unmeasured
Computational Complexity	Low	High

Table 1.1: Comparison of advantages and disadvantages of data-driven and first-principles process models.

1.1.4 On-line Estimation and Data Reconciliation

Nowadays, the development of data-driven process models can be done systematically through the design and analysis of process dynamic responses [101]. The development and implementation of first-principles models can be, on the other hand, a time-consuming and expensive task. If the process model is not currently available from simulation libraries, the model needs to be developed from the scratch using high-quality laboratory or pilot plant data.

For the implementation of a first-principles model in on-line environments, process measurements are reconciled to the model in order to close the material and energy balances and to filter out measurement errors through a *data reconciliation* task. In addition, artificial disturbances or parameters are added to the model and tuned on-line in order to account for unmodelled time-varying phenomena (e.g. catalyst deactivation, coking, fouling) encountered in industrial environments. In RTO, on-line model tuning is normally known as *on-line parameter estimation* [87]. Both the data reconciliation and the estimation problem are solved using measurements at the current steady-state operating point. For dynamic models, the on-line model tuning task is known as *on-line state estimation* [107, 104]. Here, we use a trajectory of multiple measurements to infer the current process state and disturbances. As with NMPC, the use of first-principles models is an important obstacle blocking the widespread use of on-line estimation technology.

1.2 Emerging Practices in Hierarchical Process Operations

Note that the dominating decision-making hierarchy is top-down and *one-directional*. One-directional decisions imply that the higher-levels do not exploit the lower-level degrees of freedom. In addition, the models and objectives at each level are drastically different. For instance, in planning/scheduling the objective is to maximize future profit (no process model), in RTO is to maximize current profit (steady-state first-principles model), and in MPC is minimize transition time (data-driven model). This implies that there exist significant losses of information as we move down the hierarchy. In other words, there is a *weak integration* among the decision-making layers.

The vertical integration of the decision-making hierarchy is an important problem that is attracting increased attention both from industry and academia. Many researchers are exploring formulations and algorithms to integrate tasks at different levels such as planning and scheduling [54], scheduling and target setting [24], target setting and control [65], and scheduling and control [94, 115], among others.

Consider the traditional interaction between RTO and MPC. First of all, a fundamental limitation of RTO is that it has to wait until the process is at steady-state. In addition, it does not make use of the process inputs or dynamic degrees of freedom. To overcome this, researchers have considered the possibility of adding an economic objective function directly in the MPC controller formulation [117]. While important benefits can be obtained with these formulations, this approach is limited by the fact that MPC uses an input-output data-driven model. In other words, the MPC objective can only be translated into simplified metrics in terms of measured variables such as unit throughput and reboiler duty. These metrics are inconsistent with the much more general RTO objective. In addition, a typical problem encountered in industrial RTO practice is that the targets computed through the first-principles steady-state model might become infeasible to the MPC controller or they might even have opposite steady-state gains [92, 118]. This scenario is illustrated in Fig-

ure 1.2. As a conclusion, *model inconsistency leads to a weak hierarchical integration and profitability losses.*

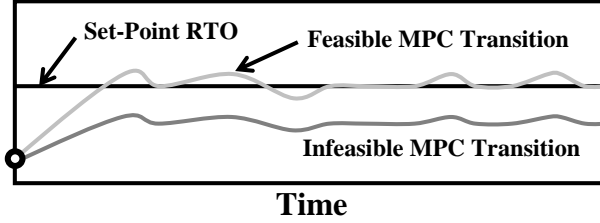


Figure 1.2: Inconsistency of RTO and data-driven MPC controller.

The possibility of using consistent models in NMPC and RTO has motivated the emerging paradigm of dynamic *real-time* optimization (D-RTO) sketched in Figure 1.3 [58, 83, 118]. Many powerful alternatives to integrate target setting and control layers can be derived using these concepts. For instance, one could consider using NMPC with an RTO economic objective to optimize the transitions between RTO steady states, or one could even consider merging both layers into a centralized decision-making layer. Finally, the ability to make economic decisions through the controller allows us to *quantify* the potential benefits of incorporating sophisticated model-based control technology in operations [39].

1.3 Research Problem Statement and Scope of Work

As we have seen, incorporating first-principles dynamic models in on-line operations can lead to a tighter integration of the decision-making hierarchy and, consequently, to an increased process performance. However, a crucial problem blocking this practice is the *computational complexity* of the associated optimization problems. In particular, solution time becomes an important issue in on-line tasks since it limits the ability to provide fast feedback to the process.

The objective of this dissertation is to identify computational bottlenecks arising in the

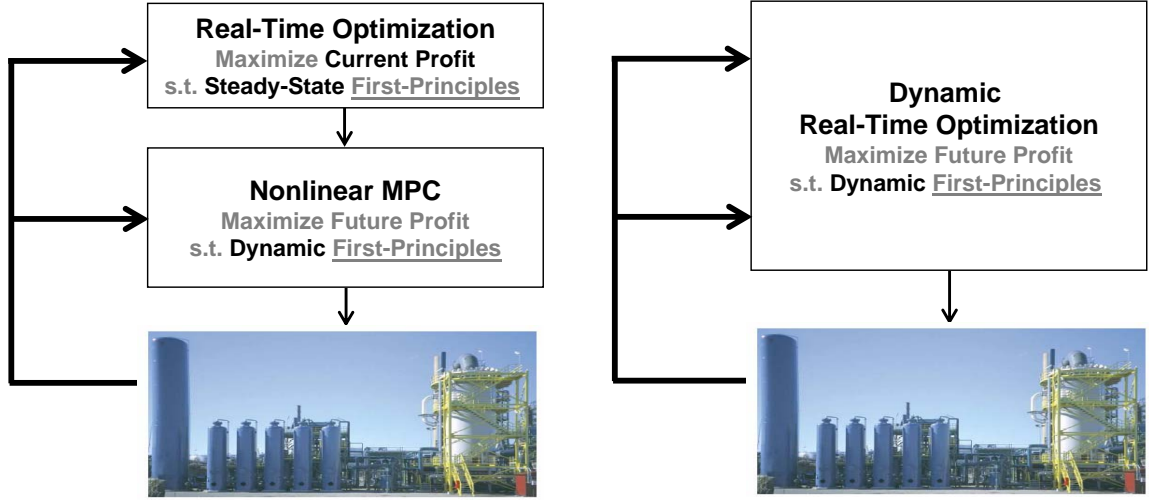


Figure 1.3: Dynamic Real-Time Optimization (D-RTO).

context of model-based process operations, and to propose strategies to overcome them. We consider specific tasks such as parameter estimation, data reconciliation, state estimation, nonlinear model predictive control, and real-time optimization. In addition, we seek to establish connections between computational and theoretical aspects of these tasks. Finally, we demonstrate the potential benefits of the developments through realistic scenarios arising in the operation of industrial processes.

1.4 Thesis Outline

This dissertation is organized in three parts. In the **first part**, we derive a computational framework for DAE-constrained optimization. The framework makes use of a simultaneous full-discretization approach to transform general DAE-constrained optimization problems into large-scale and sparse NLP problems. The advantages and disadvantages of this approach are contrasted against those of competing approaches in Chapter 2. In Chapter 3,

we discuss the use of full-space interior-point (IP) solvers for the solution of the resulting NLPs. We extend the capabilities of these solvers through a set of post-optimal analysis strategies. In particular, we present NLP sensitivity strategies to compute fast approximate solutions for neighboring problems and strategies to extract reduced Hessian information from the KKT matrix.

In the **second part**, we discuss the application of the proposed computational strategies to tackle issues arising in model-based operations. In Chapter 4 we discuss the decomposition of large multi-set parameter estimation problems and present strategies to extract parameter covariance information from the NLP solver. In Chapter 5 we discuss the role of feedback delays on the stability of NMPC controllers and derive on-line synchronization strategies based on NLP sensitivity to overcome them. In Chapter 6 we derive on-line synchronization strategies for MHE. In Chapter 7 we discuss issues arising from the coupling of on-line synchronization strategies for NMPC and MHE.

The **third part** includes Chapter 8 in which we apply the proposed developments to industrial low-density polyethylene (LDPE) tubular reactors. The dissertation closes in Chapter 9 where general concluding remarks and recommendations for future work are presented.

Part I

Computational Framework

Chapter 2

DAE-Constrained Optimization

In this chapter, we discuss advantages and disadvantages of different approaches to the solution of optimization problems constrained by differential and algebraic equations (DAEs). We make special emphasis on the computational advantages of the simultaneous full-discretization approach. In addition, we discuss advantages of the orthogonal collocation on finite elements discretization scheme.

2.1 Problem Formulation

In this dissertation, we consider general DAE-constrained optimization problems of the form,

$$\min_{u(t), p, z_0} \quad \Phi := \varphi(z_d(t_f)) \quad (2.1a)$$

$$\text{s.t.} \quad \frac{dz_d}{dt} = f_d(z_d(t), z_a(t), u(t), p, \eta) \quad (2.1b)$$

$$0 = f_a(z_d(t), z_a(t), u(t), p, \eta) \quad (2.1c)$$

$$z_d(0) = \bar{z}_0 \quad (2.1d)$$

$$z_d^L \leq z_d(t) \leq z_d^U \quad (2.1e)$$

$$z_a^L \leq z_a(t) \leq z_a^U \quad (2.1f)$$

$$u^L \leq u(t) \leq u^U \quad (2.1g)$$

$$p^L \leq p \leq p^U \quad (2.1h)$$

where t is the scalar independent dimension defined in the fixed domain $[0, t_f]$, $z_d \in \mathbb{R}^{n_d}$ is a vector of differential variables or *states*, $z_a \in \mathbb{R}^{n_a}$ is a vector of algebraic states, $u(t) \in \mathbb{R}^{n_u}$ are the inputs or *controls*, $p \in \mathbb{R}^{n_p}$ are *parameters* and η represents fixed problem *data*. Vector $f_d(\cdot) : \mathbb{R}^{n_d+n_a+n_u+n_p} \rightarrow \mathbb{R}^{n_d}$ denotes the differential equations, $f_a(\cdot) :$

$\mathbb{R}^{n_d+n_a+n_u+n_p} \rightarrow \mathbb{R}^{n_a}$ denotes the algebraic equations and $h(\cdot) : \mathbb{R}^{n_d+n_a+n_u+n_p} \rightarrow \mathbb{R}^{n_h}$ are general inequality constraints. The initial conditions of the differential states are given by \bar{z}_0 . For simplicity in the notation, the objective function $\varphi(\cdot) : \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ is assumed to be of *Mayer*-type. Nevertheless, note that any integral or *Bolza*-type objective function of the form,

$$\Phi := \int_0^{t_f} \varphi(z_d(t), z_a(t), u(t), p, \eta) dt \quad (2.2)$$

can be reformulated to Mayer form by adding artificial state variables [27].

In chemical engineering applications the DAEs (2.1b)-(2.1c) represent the system *model*. The differential equations represent conservation equations (i.e. material, energy and momentum) and the algebraic equations represent constitutive relations to compute physical, thermodynamic and transport properties. The general formulation (2.1) can represent different optimization problems arising in the context of chemical process operations such as optimal control, parameter estimation, and state estimation. For instance, in optimal control problems the degrees of freedom are the inputs and the objective is least-squares or economic, in parameter estimation problems the degrees of freedom are the parameters and inputs with a least-squares objective, in state estimation the degrees of freedom are the parameters, inputs and initial states and the objective is least-squares. Finally, note that if the system is described by partial differential, ordinary differential and algebraic equations (PDAEs), we can discretize along the spatial dimensions to recover form (2.1).

In this dissertation, we will also be interested in analyzing the effect of perturbations on the problem data η on the solution of problem (2.1). The data can be used to represent measurement information in estimation problems or the current state of the dynamic system in optimal control problems.

2.2 Approaches to DAE-Constrained Optimization

Several approaches exist to solve the DAE-constrained optimization problem (2.1). In the context of optimal control, these problems were solved until the 1970s using an *indirect* or variational approach, based on Pontryagin's Maximum Principle [21, 100]. For problems without inequality constraints, the resulting optimality conditions can be formulated and solved as a two-point boundary value problem. Indirect approaches have solid theoretical foundations based on calculus of variations. However, for problems requiring the handling of active inequality constraints, finding the correct switching structure as well as suitable initial guesses for state and adjoint variables is often very difficult. This limitation has made the indirect approach less popular in more general applications.

Direct approaches try to cast problem (2.1) as a nonlinear programming (NLP) problem. As a consequence, these approaches tend to be much more general and computationally efficient. The main fundamental difference among direct approaches is the strategy used to handle the continuous-time DAE model. In order to establish a brief comparison between these approaches, we will consider the classical optimal control problem. In this problem, we have fixed initial conditions \bar{z}_0 and parameters p and seek to find the control trajectory $u(t)$ over the time domain $[0, t_f]$ that minimizes the performance index Φ . For future reference, we will also partition the time domain $[0, t_f]$ into N *stages* where the domain inside each element k is given by $t \in [t_k, t_{k+1}]$ with $k = 0, \dots, N-1$, $t_0 = 0$ and $t_N = t_f$. Using this representation, we can pose problem (2.1) as a multi-stage problem of the form,

$$\min_{u^k(t), z_k} \quad \Phi := \sum_{k=0}^{N-1} \varphi^k(z_d^k(t_{k+1})) \quad (2.3a)$$

$$\text{s.t.} \quad \frac{dz_d^k}{dt} = f_d(z_d^k(t), z_a^k(t), u^k(t), p, \eta) \quad (2.3b)$$

$$0 = f_a(z_d^k(t), z_a^k(t), u^k(t), p, \eta) \quad (2.3c)$$

$$z_d^k(t_k) = z_k \quad (2.3d)$$

$$z_{k+1} = z_d^k(t_{k+1}) \quad (2.3e)$$

$$z_0 = \bar{z}_0 \quad (2.3f)$$

$$z_d^L \leq z_d^k(t) \leq z_d^U \quad (2.3g)$$

$$z_a^L \leq z_a^k(t) \leq z_a^U \quad (2.3h)$$

$$u^L \leq u^k(t) \leq u^U \quad (2.3i)$$

$$k = 0, \dots, N - 1.$$

Note the introduction of intermediate variables z_k that *link* the state profiles of the differential states between neighboring stages or elements. This is sketched in Figure 2.1.

2.2.1 Single-Shooting Approach

In the *single-shooting* or sequential approach, the continuous control trajectory $u(t)$ is discretized or parameterized [119, 120]. For instance, we can consider a piece-wise approximation where we fix the control $u^k(t) = u^k$ inside each element k . The discretized control trajectory can be passed to a DAE solver that integrates the DAEs over the entire domain $[0, t_f]$ in a *single* call. This fully eliminates the DAE model from the optimization problem. An NLP solver is used to obtain a new guess of the controls. If the NLP solver computes the search step for the controls using an exact Newton's method, it will require first and second-order gradient information with respect to the controls. First-order gradient information can be obtained efficiently through direct sensitivity or adjoint formulations. However, obtaining second derivative information through the DAE solver can become computationally expensive. The need for second derivatives can be avoided through Quasi-Newton approximations but this also leads to weaker convergence properties of the NLP solver. Sequential

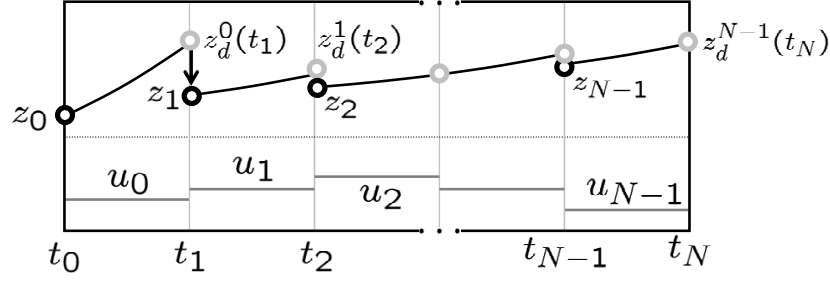


Figure 2.1: Schematic representation of multi-stage DAE-constrained optimization problem.

strategies are relatively easy to construct and to apply. Efficient implementations of the single-shooting approach incorporate highly reliable DAE solvers (e.g., DASSL, DASOLV, DAEPACK) and off-the-shelf NLP solvers as components. The use of DAE solvers allows to handle stiff dynamic systems efficiently as they can modify the integration step *adaptively* [80]. On the other hand, repeated numerical integration of the large-scale DAE model is required, which becomes time consuming in large-scale problems. The overall computational complexity of single-shooting is favorable for problems with a large number of states. However, the complexity does not scale well for problems with long time horizons and many degrees of freedom [131, 34]. Moreover, it is well known that single-shooting approaches cannot handle open loop instability [17, 45].

2.2.2 Multiple-Shooting Approach

Multiple shooting exploits the multi-stage structure of problem (2.3) [18]. Here, guesses of the parameterized controls u_k and of the initial conditions z_k are considered at each element k . Each integration is performed on the individual domain $[t_k, t_{k+1}]$ to return $z_d^k(t_{k+1})$. With this, the integration of the DAE and the evaluation of gradient information can be done separately at each element. The NLP solver updates both u_k and z_k to minimize the performance index Φ and *simultaneously* links the state profiles across elements as the problem converges. As a result of this particular construction, this approach can handle

open-loop unstable systems efficiently, which is an important advantage over single-shooting [18, 80, 34].

An important issue arising in multiple-shooting is that, when the DAE model contains many dynamic states, the gradients with respect to the states form large dense blocks inside the NLP solver [80]. Although several strategies have been proposed to handle this issue, the overall computational complexity of this approach does not scale well for problems with many dynamic states. Nevertheless, the complexity of multiple-shooting is very favorable for problems with long time horizons. [131, 34].

2.2.3 Simultaneous Full-Discretization Approach

The simultaneous full-discretization approach approximates the continuous-time profiles of the states and controls at each element k through numerical quadrature schemes [14, 29, 27, 16]. With this, the discretized DAE model can be added directly as algebraic constraints in the NLP formulation. The NLP solver simultaneously solves the DAE model as it minimizes the performance index. In other words, the discretized DAE system is solved only once, at the optimal point, and therefore avoids repeated integrations of the large-scale DAE model. Another consequence of this is that the approach can handle open-loop unstable systems. Finally, since the NLP is in completely algebraic form, first and second order derivative information can be computed cheaply through modeling platforms or automatic differentiation routines.

A problem that arises in the full-discretization approach is that a large number of discretization points and mesh refinement techniques might be required to handle stiff dynamic systems [13, 17, 80]. On the other hand, the ability to exploit the sparsity of the DAE model directly through the NLP solver has been identified as a crucial advantage of this approach over single and multiple shooting [131]. The overall computational complexity of this approach is favorable for problems with a large number of states and degrees of freedom. In

this dissertation, we will see that this favorable computational complexity becomes advantageous, particularly in on-line applications. Consequently, we will focus on the specifics of this approach. In the following section, we present details of a particular strategy for the discretization of DAE-constrained optimization problems.

Orthogonal Collocation on Finite Elements

Orthogonal collocation on finite elements is a widely used discretization scheme for the solution of DAE-constrained optimization problems. This is mainly due to its high accuracy and numerical stability properties, which make it particularly attractive for handling stiff dynamic systems [16]. In the context of optimal control, important convergence properties have also been established between this discretization scheme and the optimality conditions arising in variational approaches [106, 66].

To explain the mechanics of the orthogonal collocation scheme, we consider the DAE system of problem (2.3) defined at element k ,

$$\frac{dz_d^k}{dt} = f_d(z_d^k(t), z_a^k(t), u^k(t), p, \eta) \quad (2.4a)$$

$$0 = f_a(z_d^k(t), z_a^k(t), u^k(t), p, \eta) \quad (2.4b)$$

$$z_d^k(t_k) = z_k. \quad (2.4c)$$

It is possible to make an affine transformation of the time domain $t \in [t_k, t_{k+1}]$ at each element to $\tau = [0, 1]$ by defining $t = h_k \tau$ with $h_k = t_{k+1} - t_k$. With this,

$$\frac{dz_d^k}{d\tau} = h_k f_d(z_d^k(\tau), z_a^k(\tau), u^k(\tau), p, \eta) \quad (2.5a)$$

$$0 = f_a(z_d^k(\tau), z_a^k(\tau), u^k(\tau), p, \eta) \quad (2.5b)$$

$$z_d^k(0) = z_k. \quad (2.5c)$$

In an orthogonal collocation scheme, we approximate the state profiles using a family of polynomials defined at a set of quadrature points τ_j , $j = 1, \dots, N_c$ on the domain $\tau = [0, 1]$.

In other words,

$$\tilde{z}_d^k(\tau) \approx z_d^k(\tau) \quad (2.6a)$$

$$= \sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau) \quad (2.6b)$$

where $\tilde{z}_d^k(\tau)$ is the approximate profile, $z_d^{k,j} = z_d^k(\tau_j)$ are interpolation points and $\mathbf{\Omega}_j(\tau)$ are support or basis functions (polynomials in this case). The term *collocation* arises from the fact that we choose the family of basis functions $\mathbf{\Omega}_j(\tau)$, $j = 1, \dots, N_c$ in such a way that the approximate profile $\tilde{z}_d^k(\tau)$ is collocated or *accommodated* in order to *exactly match* the true profile at a set of points $\tau_i, j = 1, \dots, N_c$. That is,

$$\tilde{z}_d^k(\tau_i) = \sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau_i) \quad (2.7a)$$

$$= z_d^k(\tau_i), \quad i = 1, \dots, N_c. \quad (2.7b)$$

Lagrange interpolation polynomials are particularly useful to derive collocation schemes.

These polynomials have the form,

$$\mathbf{\Omega}_j(\tau) = \prod_{k=1, k \neq j}^{N_c} \frac{\tau - \tau_k}{\tau_j - \tau_k} = \frac{\tau - \tau_1}{\tau_j - \tau_1} \frac{\tau - \tau_2}{\tau_j - \tau_2} \dots \frac{\tau - \tau_{N_c}}{\tau_j - \tau_{N_c}}. \quad (2.8)$$

Note that these polynomials are of order $N_c - 1$ and satisfy,

$$\mathbf{\Omega}_j(\tau_i) = \begin{cases} 1, & \tau_i = \tau_j \\ 0, & \tau_i \neq \tau_j \end{cases} \quad (2.9)$$

with this, the approximate profile constructed from Lagrange basis function satisfies the collocation requirement (2.7).

The position of the collocation points τ_j is not arbitrary. These are normally chosen to minimize the integral of the *residuals* between the approximate and the true profile along the domain $\tau = [0, 1]$ [44]. The term *orthogonal* comes from the fact that the collocation points are chosen as the roots of orthogonal polynomials. For instance, the roots of a Legendre

polynomial are the quadrature points of a Gauss-Legendre quadrature scheme [30]. A K -th degree Legendre polynomial is given by the formula [85],

$$P_K(\tau) = \frac{1}{2^K K!} \frac{d^K}{d\tau^K} [(\tau^2 - 1)^K]. \quad (2.10)$$

These polynomials are orthogonal on the domain $[-1, 1]$. However, we are interested on the domain $[0, 1]$. We can shift the Legendre polynomials through the affine transformation $\tau \leftarrow 2\tau - 1$ to define the family of shifted Legendre polynomials $\tilde{P}_K(\tau)$ [62],

$$\tilde{P}_K(\tau) = \frac{1}{K!} \frac{d^K}{d\tau^K} [(\tau^2 - \tau)^K]. \quad (2.11)$$

The roots of these polynomials provide the quadrature points $\tau_j, j = 1, \dots, N_c$ with $K = N_c$. Gauss-Legendre provides the highest accuracy among collocation methods. However, a practical problem arising in this scheme is that the collocation points lie strictly in the interior of the domain $\tau \in (0, 1)$. Because of this, an extrapolation step is required to match the state profiles of the current element k with those of the neighboring elements $k - 1$ and $k + 1$. This becomes an important issue in the context of DAE-constrained optimization where flexibility is often desired in order to impose constraints at the boundary of the elements [16]. The Gauss-Legendre-Radau (Radau) quadrature scheme is a practical approach to overcome some of these limitations. Here, we impose the constraint that the last collocation point should lie at the boundary of the domain ($\tau = 1$) [30, 16]. Since the collocation points are chosen with one less degree of freedom, Radau quadrature is less accurate than Gauss quadrature. In a Radau scheme, the collocation points are the roots of the polynomial,

$$\mathbf{P}_K = \tilde{P}_K(\tau) - \tilde{P}_{K-1}(\tau). \quad (2.12)$$

Here, $\tau_j > 0$, $\tau_{N_c} = 1$ and $K = N_c$. Having the quadrature scheme, we proceed to transform the continuous-time DAE system to algebraic form. First note that we can discretize the

algebraic states and controls profiles at the same points as the differential states,

$$\tilde{z}_a^k(\tau) \approx z_a^k(\tau) = \sum_{j=1}^{N_c} z_a^{k,j} \mathbf{\Omega}_j(\tau) \quad (2.13a)$$

$$\tilde{u}^k(\tau) \approx u^k(\tau) = \sum_{j=1}^{N_c} u^{k,j} \mathbf{\Omega}_j(\tau). \quad (2.13b)$$

In addition, note that the derivative of the approximate differential state (2.6) with respect to time becomes,

$$\frac{d\tilde{z}_d^k(\tau)}{d\tau} = \sum_{j=1}^{N_c} z_d^{k,j} \dot{\mathbf{\Omega}}_j(\tau). \quad (2.14)$$

Substituting these expressions on the DAE system (2.5) we obtain,

$$\sum_{j=1}^{N_c} z_d^{k,j} \dot{\mathbf{\Omega}}_j(\tau) = h_k f_d \left(\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau), \sum_{j=1}^{N_c} z_a^{k,j} \mathbf{\Omega}_j(\tau), \sum_{j=1}^{N_c} u^{k,j} \mathbf{\Omega}_j(\tau), p, \eta \right) \quad (2.15a)$$

$$0 = f_a \left(\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau), \sum_{j=1}^{N_c} z_a^{k,j} \mathbf{\Omega}_j(\tau), \sum_{j=1}^{N_c} u^{k,j} \mathbf{\Omega}_j(\tau), p, \eta \right) \quad (2.15b)$$

$$\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(0) = z_k. \quad (2.15c)$$

In order to solve the discretized DAE system for a fixed set of inputs $u^{k,j}$, we need to find the unknown values $z_d^{k,j}, z_a^{k,j}$. This requires $N_c(n_d+n_a)$ equations. The required expressions can be obtained by enforcing (2.5) at the collocation points $\tau_i, i = 1, \dots, N_c$. With this,

$$\sum_{j=1}^{N_c} z_d^{k,j} \dot{\mathbf{\Omega}}_j(\tau_i) = h_k f_d \left(\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau_i), \sum_{j=1}^{N_c} z_a^{k,j} \mathbf{\Omega}_j(\tau_i), \sum_{j=1}^{N_c} u^{k,j} \mathbf{\Omega}_j(\tau_i), p, \eta \right) \quad (2.16a)$$

$$0 = f_a \left(\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(\tau_i), \sum_{j=1}^{N_c} z_a^{k,j} \mathbf{\Omega}_j(\tau_i), \sum_{j=1}^{N_c} u^{k,j} \mathbf{\Omega}_j(\tau_i), p, \eta \right) \quad (2.16b)$$

$$\sum_{j=1}^{N_c} z_d^{k,j} \mathbf{\Omega}_j(0) = z_k. \quad (2.16c)$$

$$i = 1, \dots, N_c.$$

Since, by construction, the approximate profiles satisfy the collocation property,

$$\sum_{j=1}^{N_c} z_d^{k,j} \Omega_j(\tau_i) \equiv z_d^{k,i}, \quad \sum_{j=1}^{N_c} z_a^{k,j} \Omega_j(\tau_i) \equiv z_a^{k,i}, \quad \sum_{j=1}^{N_c} u^{k,j} \Omega_j(\tau_i) \equiv u^{k,i} \quad (2.17a)$$

we obtain,

$$\sum_{j=1}^{N_c} z_d^{k,j} \dot{\Omega}_j(\tau_i) = h_k f_d \left(z_d^{k,i}, z_a^{k,i}, u^{k,i}, p, \eta \right) \quad (2.18a)$$

$$0 = f_a \left(z_d^{k,i}, z_a^{k,i}, u^{k,i}, p, \eta \right) \quad (2.18b)$$

$$\sum_{j=1}^{N_c} z_d^{k,j} \Omega_j(0) = z_k. \quad (2.18c)$$

$$i = 1, \dots, N_c.$$

Finally, we can couple the discretized DAEs for all the finite elements. For this, we recall that the last Radau collocation point is at the right boundary of each element. Consequently, $z_k = z_d^{k-1, N_c}, k = 1, \dots, N$ and $z_0 = \bar{z}_0$. Nevertheless, note that we still need the extrapolation step (2.18c) to match the initial conditions on the left hand side of the boundary.

An interesting family of discretization schemes results from the so-called monomial basis representation [6, 16]. In particular, it has been found that orthogonal collocation on finite elements schemes correspond to a particular class of implicit Runge Kutta methods [5]. For instance, if we define an approximate profile of the *derivative* of the differential states as,

$$\frac{d\tilde{z}_d^k(\tau)}{d\tau} = \sum_{j=1}^{N_c} \frac{dz_d^{k,j}}{d\tau} \hat{\Omega}_j(\tau) \approx \frac{dz_d^k(\tau)}{d\tau} \quad (2.19)$$

where $\hat{\Omega}_j(\tau)$ is a Lagrange polynomial of order N_c defined at the Radau collocation points $\tau_j, j = 1, \dots, N_c$. Assume that the approximate derivative profile (2.19) satisfies the following properties,

$$\tilde{z}_d^k(0) \equiv z_k \quad (2.20a)$$

$$\frac{d\tilde{z}_d^k(\tau_i)}{d\tau} \equiv \frac{dz_d^k(\tau_i)}{d\tau} = \frac{dz_d^{k,i}}{d\tau}, \quad i = 1, \dots, N_c. \quad (2.20b)$$

With this, we can establish,

$$\int_0^\tau \frac{dz_d^k(\bar{\tau})}{d\bar{\tau}} d\bar{\tau} = \int_0^\tau \sum_{j=1}^{N_c} \frac{dz_d^{k,j}}{d\bar{\tau}} \hat{\Omega}_j(\bar{\tau}) d\bar{\tau} \quad (2.21a)$$

$$z_d^k(\tau) - z_k = \sum_{j=1}^{N_c} \frac{dz_d^{k,j}}{d\tau} \int_0^\tau \hat{\Omega}_j(\bar{\tau}) d\bar{\tau} \quad (2.21b)$$

$$z_d^k(\tau) = z_k + \sum_{j=1}^{N_c} \frac{dz_d^{k,j}}{d\tau} \bar{\Omega}_j(\tau) \quad (2.21c)$$

where $\bar{\Omega}_j(\tau) := \int_0^\tau \hat{\Omega}_j(\bar{\tau}) d\bar{\tau}$ is a polynomial of order $N_c + 1$. Note that the collocation conditions (2.20) require this polynomial to satisfy,

$$\bar{\Omega}_j(0) = 0 \quad (2.22a)$$

$$\dot{\bar{\Omega}}_j(\tau_i) = \hat{\Omega}_j(\tau_i) = \begin{cases} 1, & \tau_i = \tau_j \\ 0, & \tau_i \neq \tau_j \end{cases} \quad (2.22b)$$

so we can establish,

$$z_d^{k,i} = z_k + \sum_{j=1}^{N_c} \bar{\Omega}_j(\tau_i) \frac{dz_d^{k,j}}{d\tau}, \quad i = 1, \dots, N_c \quad (2.23)$$

Finally, we define approximate profiles for the algebraic states and controls as,

$$\tilde{z}_a^k(\tau) = \sum_{j=1}^{N_c} z_a^{k,j} \hat{\Omega}_j(\tau) \quad (2.24a)$$

$$\tilde{u}^k(\tau) = \sum_{j=1}^{N_c} u^{k,j} \hat{\Omega}_j(\tau). \quad (2.24b)$$

where $\hat{\Omega}_j(\tau)$ is a Lagrange polynomial of order N_c . With this, we can express the discretized DAE system over the entire time domain $[0, t_f]$ as,

$$z_d^{k,i} = z_k + h_k \sum_{j=1}^{N_c} \bar{\Omega}_j(\tau_i) f_d(z_d^{k,j}, z_a^{k,j}, u^{k,j}, p, \eta) \quad (2.25a)$$

$$0 = f_a(z_d^{k,i}, z_a^{k,i}, u^{k,i}, p, \eta) \quad (2.25b)$$

$$z_{k+1} = z_d^{k,N_c}, \quad k = 0, \dots, N-1, \quad i = 1, \dots, N_c \quad (2.25c)$$

$$z_0 = \bar{z}_0 \quad (2.25d)$$

with $h_k = t_{k+1} - t_k$, $k = 0, \dots, N-1$ and $t_N = t_f$. It has been observed in practice that a monomial basis representation of the collocation equations gives better conditions numbers of the interpolation matrix. This is particularly relevant in large-scale applications [16]. In addition, the monomial basis representation allows to analyze the properties of collocation schemes based on widely studied properties of Runge-Kutta methods [5].

Having the discretized DAE model, we can express the DAE-constrained optimization problem (2.3) as an NLP of the form,

$$\min_{z_d^{k,i}, z_a^{k,i}, u^{k,i}, z_k} \quad \Phi := \sum_{k=0}^{N-1} \varphi^k(z_d^{k,N_c}) \quad (2.26a)$$

$$\text{s.t. } z_d^{k,i} = z_k + h_k \sum_{j=1}^{N_c} \bar{\Omega}_j(\tau_i) f_d(z_d^{k,j}, z_a^{k,j}, u^{k,j}, p, \eta) \quad (2.26b)$$

$$0 = f_a(z_d^{k,i}, z_a^{k,i}, u^{k,i}, p, \eta) \quad (2.26c)$$

$$z_{k+1} = z_d^{k,N_c} \quad (2.26d)$$

$$z_0 = \bar{z}_0 \quad (2.26e)$$

$$z_d^L \leq z_d^{k,i} \leq z_d^U \quad (2.26f)$$

$$z_a^L \leq z_a^{k,i} \leq z_a^U \quad (2.26g)$$

$$u^L \leq u^{k,i} \leq u^U \quad (2.26h)$$

$$k = 0, \dots, N-1, i = 1, \dots, N_c.$$

Note that the above formulation assumes that one finite element is placed at each stage. This can be easily generalized to allow for multiple finite elements per stage. Also note that the NLP resulting from the full-discretization approach is a large-scale problem but it is also very sparse due to the incorporation of the DAE model as constraints and due to the structure induced by the finite element scheme. Nevertheless, note that since all the complexity of the DAE model is now handled by the NLP solver, efficient solvers are required.

2.3 Concluding Remarks

In this chapter, we have summarized advantages and disadvantages of different approaches to DAE-constrained optimization. In particular, we have emphasized that the ability to avoid repetitive DAE integrations and the ability to exploit the sparsity of the DAE model directly in the NLP solver are some of the most important advantages of the full-discretization approach. In addition, we have described the use of orthogonal collocation on finite elements discretization schemes to convert general DAE-constrained optimization problem into large-scale and sparse NLPs. In the following chapter, we will present the specifics of an interior-point solver able to exploit the sparsity of these NLPs and able to handle highly complex DAE models.

Chapter 3

Interior-Point NLP Solvers

In this chapter, we discuss advantages and disadvantages of interior-point solvers for the solution of large-scale NLPs arising in the context of DAE-constrained optimization. We then discuss the implementation of post-optimal analysis capabilities on the IPOPT solver. In particular, we derive strategies to compute fast approximate solutions for neighboring problems using NLP sensitivity. In the following chapters, we will see that we can use these approximations to minimize the on-line solution time of NMPC and MHE problems. In addition, we derive strategies to extract reduced Hessian information from the NLP solver. These capabilities will be later used to extract covariance information from large-scale parameter and state estimation problems.

3.1 IPOPT Algorithm

The NLP problem (2.26) can be posed in the general form,

$$\mathcal{N}(\eta) \min_x f(x, \eta) \tag{3.1a}$$

$$s.t. \ c(x, \eta) = 0 \tag{3.1b}$$

$$x \geq 0 \tag{3.1c}$$

where $x \in \mathbb{R}^{n_x}$ is variable vector containing all the discretized states, controls and parameters, and $\eta \in \mathbb{R}^{n_\eta}$ is a *fixed* data vector.

Interior-point solvers have become a popular choice for the solution of large-scale and sparse NLPs. In particular, the solvers LOQO, KNITRO and IPOPT are widely used. In

this work, we describe some of the specifics of IPOPT, an NLP solver originally developed in our research group [121]. In interior-point approaches, the inequality constraints of problem (3.1) are handled *implicitly* by adding barrier terms to the objective function,

$$\min_x \phi(x, \eta, \mu) := f(x, \eta) + \mu \sum_{j=1}^{n_x} \ln(x_{(j)}), \quad \text{s.t.} \quad c(x, \eta) = 0 \quad (3.2)$$

where $x_{(j)}$ denotes the j th component of vector x . Solving (3.2) for a decaying sequence of $\mu \rightarrow 0$ results in an efficient strategy to solve the original NLP (3.1). The barrier approach avoids the combinatorial complexity of choosing the right active-set, which is particularly useful for NLPs with many inequality constraints. Using an initial barrier parameter μ , IPOPT follows a primal-dual approach¹ to solve the Karush-Kuhn-Tucker (KKT) conditions of the barrier problem (3.2),

$$r_x := \nabla_x \mathcal{L} = \nabla_x f(x, \eta) + \nabla_x c(x, \eta)^T \lambda - \nu = 0 \quad (3.3a)$$

$$r_\lambda := \nabla_\lambda \mathcal{L} = c(x, \eta) = 0 \quad (3.3b)$$

$$r_\nu := \mathbf{X} \mathbf{V} e - \mu e = 0 \quad (3.3c)$$

where $\mathbf{X} = \text{diag}(x)$, $\mathbf{V} = \text{diag}(\nu)$ and $e \in \mathbb{R}^{n_x}$ is a vector of ones. Symbols $\lambda \in \mathbb{R}^{n_\lambda}$ and $\nu \in \mathbb{R}^{n_x}$ are Lagrange multipliers for the equality constraints and bounds, respectively. Note that this system of nonlinear equations is parameterized by μ . Solutions of this system for decaying values of μ form the so-called *central path*. For a fixed value μ_ℓ , IPOPT applies an exact Newton's method with the iteration sequence initialized at $s_o^T := [x_o^T \ \lambda_o^T \ \nu_o^T]$. At the i th iteration, the search direction $\Delta s_i = s_{i+1} - s_i$ is computed by linearization of the KKT conditions (3.3),

$$\begin{bmatrix} \mathbf{H}_i & \mathbf{A}_i^T & -\mathbb{I}_{n_x} \\ \mathbf{A}_i & 0 & 0 \\ \mathbf{V}_i & 0 & \mathbf{X}_i \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta \lambda_i \\ \Delta \nu_i \end{bmatrix} = - \begin{bmatrix} \nabla_x f(x_i, \eta) + \mathbf{A}_i^T \lambda_i - \nu_i \\ c(x_i, \eta) \\ \mathbf{X}_i \mathbf{V}_i e - \mu_\ell e \end{bmatrix} \quad (3.4)$$

¹The term *primal-dual* refers to the fact that *artificial* dual variables ν are introduced to reformulate the KKT conditions of the barrier problem.

where $\mathbf{A}_i := \nabla_x c(x_i, \eta) \in \Re^{n_\lambda \times n_x}$ is the constraint Jacobian and $\mathbf{H}_i := \nabla_{xx} \mathcal{L} \in \Re^{n_x \times n_x}$ is the Hessian of the Lagrange function,

$$\mathcal{L} = f(x_i, \eta) + \lambda_i^T c(x_i, \eta) - \nu_i^T x_i, \quad (3.5)$$

and \mathbb{I}_{n_x} denotes the identity matrix. The matrix on the left-hand side of (3.4) is known as the *primal-dual matrix* and we will denote this as $\bar{\mathbf{K}}_i$.

In IPOPT, we can provide exact Hessian and Jacobian information through the modeling platform AMPL [47]. With this, it is possible to guarantee fast local convergence of Newton's method, and it is possible to handle problems with many degrees of freedom without altering these local convergence properties (as opposed to Quasi-Newton methods). After solving a sequence of barrier problems for $\mu_\ell \rightarrow 0$, the solver returns the optimal solution triplet $s_*^T = [x_*^T \ \lambda_*^T \ \nu_*^T]$ which implicitly defines the *active-set* (set of variables satisfying $x_{(j)} = 0$). To update the barrier parameter μ_ℓ , IPOPT can use a monotonic decrease strategy or an adaptive decrease strategy. It is worth emphasizing that these strategies do not necessarily follow the central path (as opposed to path-following approaches such as Mehrotra's predictor-corrector method). Finally, the solver incorporates a filter line-search globalization strategy to promote convergence from poor starting points.

Remark: The primal-dual system (3.4) is a simplification of the actual system implemented in the solver which incorporates general inequality constraints (see Appendix A). This is done in order to simplify the presentation.

3.1.1 Computational Issues

The primal-dual system is symmetrized by eliminating the step for the dual variables. This gives,

$$\begin{bmatrix} \mathbf{W}_i + \delta_x \mathbb{I}_{n_x} & \mathbf{A}_i^T \\ \mathbf{A}_i & -\delta_c \mathbb{I}_{n_\lambda} \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta \lambda_i \end{bmatrix} = - \begin{bmatrix} \nabla_x \bar{\mathcal{L}}(x_i, \lambda_i, \eta) \\ c(x_i, \eta) \end{bmatrix}, \quad (3.6a)$$

$$\Delta \nu_i = \mathbf{X}_i^{-1} (\mu_\ell e - \mathbf{V}_i \Delta x_i) - \nu_i \quad (3.6b)$$

where $\mathbf{W}_i := \mathbf{H}_i + \mathbf{\Sigma}_i$ is known as the *barrier Hessian*, $\nabla_x \bar{\mathcal{L}}(x_i, \lambda_i, \eta) := \nabla_x \phi(x_i, \eta, \mu_\ell) + \mathbf{A}_i^T \lambda_i$, $\mathbf{\Sigma}_i := \mathbf{X}_i^{-1} \mathbf{V}_i$ and $\delta_x, \delta_c \geq 0$. The matrix on the left-hand side of (3.6a) is known as the *augmented matrix*. We will also refer to this matrix as the KKT² matrix and denote this as \mathbf{K}_i . This is a large-scale, sparse, symmetric, and indefinite matrix. The computation of the search step involves the factorization of this augmented matrix. This factorization is the most computationally intensive step in the solution of the NLP. A crucial advantage that interior-point solvers offer over active-set solvers is that the structure of the primal-dual matrix and of the KKT matrix *do not change* between iterations. This facilitates the design of tailored linear algebra strategies to exploit special structures.

In IPOPT, we use a direct factorization of the augmented matrix as a default. With this, we only exploit its overall *sparsity pattern*. The computational complexity of this strategy is in general very favorable, scaling nearly linearly, and at most quadratically, with the overall dimensions of the NLP [131]. This is a significant advantage in the context of DAE-constrained optimization since, as we have seen, the NLP tends to be very sparse. However, a problem that might arise in the direct factorization approach is that significant fill-in and computer memory bottlenecks can be introduced during the factorization step if the sparsity pattern is not properly exploited. In order to factorize the KKT matrix, we can use several sparse linear solvers interfaced to IPOPT such as MA27, MA57, Pardiso, among others [110, 37]. These linear solvers perform a preliminary analysis of the matrix sparsity pattern. During this phase, the linear solver permutes the matrix to reduce fill-in and computer memory requirements in the factorization phase. We have recently found out that MA57 is particularly attractive since it incorporates different ordering strategies such as an approximate minimum degree (AMD) ordering algorithm and a nested dissection algorithm based on the multi-level graph partitioning strategy, implemented in Metis [67, 52]. For very large-scale problems, nested dissection techniques excel at identifying high-level (coarse-grained) structures and thus play a crucial role in the factorization time and reliability of

²In numerical linear algebra literature, the structure of the augmented matrix is said to be of KKT type.

the linear solver. This is a particularly attractive feature in the context of DAE-constrained optimization since many (sometimes non-intuitive) structures arise.

3.1.2 Inertia Correction

In order to compute the search step, the augmented matrix needs to be *non-singular*. In addition, the filter line-search strategy used in IPOPT requires that the Newton step is a descent direction [121]. These requirements might not hold in the presence of directions of negative curvature or rank-deficient Jacobians. These degeneracies tend to appear in highly nonlinear problems and/or ill-posed formulations. This is particularly relevant in the context of DAE-constrained optimization since all the complexity of the DAE model is handled by the NLP solver. IPOPT follows a special inertia-correction strategy to cope with these issues. Because of the importance of this topic, we summarize a set of results that will become useful in the following sections.

Definition 1. (*Reduced Barrier Hessian [93]*) Let $Z \in \mathbb{R}^{n_x \times n_d}$ with $n_d := n_x - n_\lambda$ be a matrix whose columns form a null-space of the Jacobian of the equality constraints \mathbf{A} (i.e. Z has full rank and $\mathbf{A}Z = 0$). The reduced barrier Hessian matrix is defined as $Z^T \mathbf{W} Z$ with \mathbf{W} defined in (3.6).

Lemma 3.1.1. (*Non-singularity of KKT matrix [46]*) Let \mathbf{A} be full row rank and assume that the reduced barrier Hessian $Z^T \mathbf{W} Z$ is positive definite then, the KKT system

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{A}^T \\ \mathbf{A} & \end{bmatrix} \quad (3.7)$$

is non-singular.

Note that the reduced barrier Hessian is never formed in full-space solvers like IPOPT. Therefore, it would be impractical to check for the non-singularity of the KKT matrix in this way. An indirect strategy can be derived from the following results.

Definition 2. (*Inertia of a Matrix [46]*) Given a symmetric matrix \mathbf{K} , its inertia, denoted by $\text{In}(\mathbf{K})$, is the integer triple (i_p, i_n, i_z) where $i_p(\mathbf{K})$, $i_n(\mathbf{K})$, and $i_z(\mathbf{K})$ are the numbers of positive, negative, and zero eigenvalues of \mathbf{K} .

Lemma 3.1.2. (*Relationship Inertia of KKT matrix and Reduced Barrier Hessian [93]*) If \mathbf{A} has rank n_λ , then $\text{In}(\mathbf{K}) = \text{In}(Z^T \mathbf{W} Z) + (n_\lambda, n_\lambda, 0)$.

If the reduced barrier Hessian is positive definite, then $\text{In}(Z^T \mathbf{W} Z) = (n_d, 0, 0)$ with $n_d = n_x - n_\lambda$. Using this fact, the above lemma leads to,

Theorem 3.1.3. (*Inertia and non-singularity of KKT matrix [93]*) Assume \mathbf{A} has rank n_λ and that the reduced Hessian $Z^T \mathbf{W} Z$ is positive definite then, $\text{In}(\mathbf{K}) = (n_x, n_\lambda, 0)$.

Since many symmetric linear solvers such as MA27 or MA57 return the inertia of the KKT matrix as an outcome of the factorization, it is possible to use this to verify that the Newton step is indeed a descent direction and/or if the Jacobian has full row rank [93]. If the inertia returned by the linear solver is not exactly $(n_x, n_\lambda, 0)$, then IPOPT refactorizes the KKT matrix (3.6a) with different trial values of the diagonal terms $\delta_x \geq 0$ and $\delta_c \geq 0$ until the required inertia is obtained [121]. Once the problem *is converged*, the inertia of the KKT matrix can give an indication of the nature of the optimal point since we now have a way to verify the status of the reduced barrier Hessian at each iteration. To explain this, we make use of the following fundamental results.

Definition 3. (*Jacobian of Active Constraints*) Let $\mathbf{A} \in \mathbb{R}^{n_\lambda \times n_x}$ be the Jacobian of the equality constraints. Consider a variable partition $x^T = [x_A^T | x_I^T]$ where $x_A \in \mathbb{R}^{n_A}$ are the active variables at the solution ($x_{A(j)} = 0$) and $x_I \in \mathbb{R}^{n_I}$ are the inactive variables ($x_{I(j)} > 0$) with $n_x = n_A + n_I$ and associated bound multipliers ν_A and ν_I . Let $\mathbf{E} \in \mathbb{R}^{n_A \times n_x}$ be the Jacobian of the active bounds with structure $\mathbf{E} = [\mathbb{I}_{n_A} | 0]$. The Jacobian of the active constraints is then defined as $\bar{\mathbf{A}}^T = [\mathbf{A}^T | \mathbf{E}^T]$ with $\bar{\mathbf{A}} \in \mathbb{R}^{(n_\lambda + n_A) \times n_x}$.

Definition 4. (*LICQ [93]*) We say that the linear independence constraint qualification holds at a point s_* if all the rows of $\bar{\mathbf{A}}$ are linearly independent.

Definition 5. (*Strict Complementarity [93]*) We say that strict complementarity holds at s_* if for all $x_{A(j)}$ then $\nu_{A(j)} > 0$ and for all $x_{I(j)}$ then $\nu_{I(j)} = 0$.

Lemma 3.1.4. (*First-Order Necessary Conditions (KKT) [93]*) Assume that s_* is a local solution of the NLP (3.1) and that, at this point, LICQ holds. Then, there exist multipliers λ_* and ν_* such that,

$$\nabla_x \mathcal{L}(s_*) = 0 \quad (3.8a)$$

$$\nabla_\lambda \mathcal{L}(s_*) = 0 \quad (3.8b)$$

$$\mathbf{X}_* \mathbf{V}_* = 0 \quad (3.8c)$$

$$x_* \geq 0 \quad (3.8d)$$

$$\nu_* \geq 0. \quad (3.8e)$$

Lemma 3.1.5. (*Second-Order Sufficient Conditions (SSOC) [93]*) Assume that s_* satisfies the KKT conditions and that, at this point, strict complementarity holds. Assume also that, for a nonzero vector $w \in \mathbb{R}^{n_x}$ satisfying $\bar{\mathbf{A}}w = 0$,

$$w^T \mathbf{H} w > 0 \quad (3.9)$$

holds, then s_* is a strict local solution of the NLP (3.1).

Note that we can always project a vector w satisfying $\bar{\mathbf{A}}w = 0$ as $w = \bar{\mathbf{Z}}u$ with $\bar{\mathbf{A}}\bar{\mathbf{Z}} = 0$. As a consequence, condition (3.9) is equivalent to say that the *reduced Hessian* $\bar{\mathbf{Z}}^T \mathbf{H} \bar{\mathbf{Z}}$ is positive definite. Note also that the *reduced Hessian* is a projection of the Lagrangian Hessian \mathbf{H} into the null-space of all the active constraints. When no bounds are active at the solution then $\mathbf{\Sigma} = 0$ and $\mathbf{A} = \bar{\mathbf{A}}$. As a consequence, the reduced barrier Hessian and the reduced Hessian coincide. Since we can check for positive definiteness of the reduced barrier Hessian through the regularization terms δ_c and δ_d , we can directly check if the SSOC qualification holds. When we have active bounds, we can also guarantee that the positive definiteness of the reduced barrier Hessian implies that the optimal point satisfies SSOC. In order to prove this, we establish the following results.

Lemma 3.1.6. (*Debreu's Lemma [46]*) Given an $n_A \times n_x$ matrix \mathbf{E} and an $n_x \times n_x$ symmetric matrix \mathbf{H} , then $w^T \mathbf{H} w > 0$ for all nonzero w satisfying $\mathbf{E} w = 0$ if and only if there is a finite $\bar{\rho} > 0$ such that $\mathbf{H} + \rho \mathbf{E}^T \mathbf{E}$ is positive definite for all $\rho \geq \bar{\rho}$.

Theorem 3.1.7. (*Reduced Barrier Hessian and SSOC*) Assume that s_* satisfies the KKT conditions and that, at this point, strict complementary holds. Assume also that, at this point, the reduced barrier Hessian is positive definite (i.e. $w^T \mathbf{W} w > 0$ for all nonzero vectors w satisfying $\mathbf{A} w = 0$) then, SSOC holds.

Proof: Consider the variable partition $x^T = [x_A^T | x_I^T]$. Recall that $\mathbf{W} = \mathbf{H} + \mathbf{\Sigma}$ with $\mathbf{\Sigma} = \mathbf{X}^{-1} \mathbf{V} = \mu \mathbf{X}^{-2}$. The barrier Hessian can be written as,

$$\mathbf{W} = \mathbf{H} + \begin{bmatrix} \rho \mathbb{I}_{n_A} & 0 \\ 0 & O(\mu) \mathbb{I}_{n_I} \end{bmatrix}. \quad (3.10)$$

As $\mu \rightarrow 0$ the block on the right corner of $\mathbf{\Sigma}$ tends to zero. On the other hand, the block on the left corner tends to a large value as $\mu \rightarrow 0$ that we bound from above through the constant $\rho > 0$. With this, we can establish,

$$w^T \mathbf{W} w = w^T \mathbf{H} w + \rho w^T \mathbf{E}^T \mathbf{E} w. \quad (3.11)$$

From Debreu's lemma we know that $w^T \mathbf{H} w > 0$ holds for all nonzero w satisfying $\mathbf{E} w = 0$ if and only if \mathbf{W} is positive definite. Furthermore, positive definiteness \mathbf{W} implies that for all nonzero w such that $\mathbf{A} w = 0$ then $w^T \mathbf{W} w > 0$. The result follows directly from Definition 3 and Lemma 3.1.5. \square

The above result is of practical importance since it implies that if the KKT matrix does not need regularization at the solution, we can guarantee that the optimal point satisfies SSOC. This is a quick and direct check of the reliability of the solution and can be helpful in determining if the NLP is *ill-posed* (e.g. convergence to saddle points).

3.1.3 IPOPT Software Architecture

IPOPT is currently implemented in object-oriented or modular form (see Figure 3.1). The structure is designed so that the main algorithmic components such as the globalization strategy, the μ update strategy, the solution of the augmented linear system, etc. can be seen as *classes*. Any particular implementation of a particular class is seen as an *object* which inherits all the properties of the class [77].

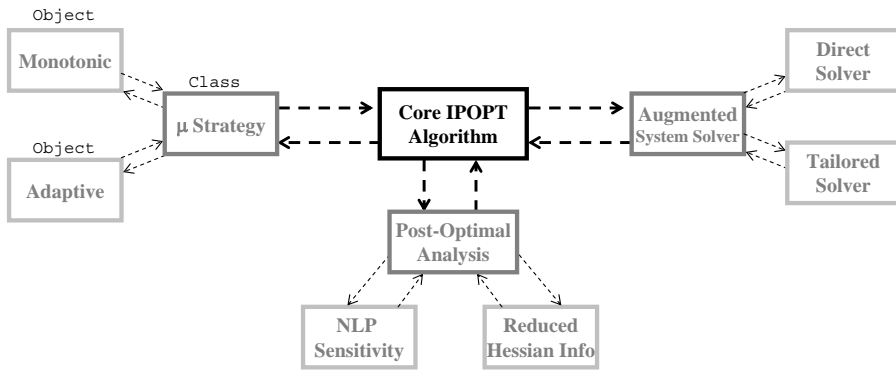


Figure 3.1: Object-Oriented Implementation of IPOPT.

An important advantage of this software architecture is that extra capabilities can be added to the solver without altering the *core* structure of the algorithm. In the following sections, we describe a set of new capabilities added to the IPOPT solver. In particular, we are interested in reusing the factorization of the augmented matrix available at the solution to perform a variety of *Post-Optimal Analysis* tasks. For instance, we can analyze the curvature of the solution by extracting reduced Hessian information and we can perform fast NLP sensitivity calculations to compute approximate solutions for neighboring NLP problems.

3.2 Post-Optimal Analysis

3.2.1 Extraction of Reduced Hessian Information

In certain applications, we would not only want to verify but also to *quantify* the curvature of the solution. This can be done by computing the principal components or eigenvalues of the reduced Hessian matrix. A problem that arises in IPOPT is that the reduced Hessian is never formed. Here, we present a practical strategy to extract this information from the KKT matrix through inexpensive backsolves.

To explain the mechanics of the strategy, we split the variable vector as $x^T = [b^T \ d^T]$ where $b \in \mathbb{R}^{n_\lambda}$ is a vector of dependent variables and $d \in \mathbb{R}^{n_d}$ is a vector of independent variables of the same dimension as the number of degrees of freedom of the NLP (i.e. $n_d = n_x - n_\lambda$). According to this variable partition, the Jacobian can be rearranged as $\mathbf{A} = [\mathbf{A}_b \mid \mathbf{A}_d]$ where $\mathbf{A}_b \in \mathbb{R}^{n_\lambda \times n_\lambda}$ is a non-singular square matrix. In addition, the barrier Hessian can be represented as,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{bb} & \mathbf{W}_{bd} \\ \mathbf{W}_{db} & \mathbf{W}_{dd} \end{bmatrix}. \quad (3.12)$$

Following this reasoning, we can represent the KKT system (3.6) *evaluated at the solution* as,

$$\begin{bmatrix} \mathbf{W}_{bb} & \mathbf{W}_{bd} & \mathbf{A}_b^T \\ \mathbf{W}_{db} & \mathbf{W}_{dd} & \mathbf{A}_d^T \\ \mathbf{A}_b & \mathbf{A}_d & 0 \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta d \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_b \\ r_d \\ r_\lambda \end{bmatrix}. \quad (3.13)$$

Here, note that the right-hand sides are zero at the solution of the NLP and that the KKT matrix is already factorized. A practical strategy to extract reduced Hessian information using this factorization is summarized in the following theorem.

Theorem 3.2.1. (*Extraction of Reduced Barrier Hessian from Factorized KKT Matrix*) Assume that the KKT matrix is non-singular at the current optimal point. If we set $r_b = r_\lambda = 0$ and $r_d = \mathbb{I}_{n_d}(:, j)$ (i.e. the j th column of the identity matrix) then, the search step Δd becomes the j th column of the inverse of the reduced barrier Hessian matrix. In other words,

$$\Delta d = (Z^T \mathbf{W} Z)^{-1}(:, j) \quad (3.14)$$

where $Z \in \mathbb{R}^{n_x \times n_d}$ is a null-space matrix of \mathbf{A} with structure,

$$Z = \begin{bmatrix} -\mathbf{A}_b^{-1} \mathbf{A}_d \\ \mathbb{I}_{n_d} \end{bmatrix}. \quad (3.15)$$

Proof: Using Z defined in (3.15) and \mathbf{W} defined in (3.13), the structure of the reduced barrier Hessian is,

$$Z^T \mathbf{W} Z = \mathbf{W}_{dd} + \mathbf{A}_d^T \mathbf{A}_b^{-T} \mathbf{W}_{bb} \mathbf{A}_b^{-1} \mathbf{A}_d - \mathbf{A}_d^T \mathbf{A}_b^{-T} \mathbf{W}_{bd} - \mathbf{W}_{db} \mathbf{A}_b^{-1} \mathbf{A}_d. \quad (3.16)$$

Note that performing backsolves with the factorized KKT matrix (3.13) and $\nabla_b \bar{\mathcal{L}} = \nabla_\lambda \bar{\mathcal{L}} = 0$ and $\nabla_d \bar{\mathcal{L}} = \mathbb{I}_{n_d}(:, j)$ is equivalent to solve the linear system,

$$\mathbf{W}_{bb} \Delta B + \mathbf{W}_{bd} \Delta D + \mathbf{A}_b^T \Delta \Lambda = 0 \quad (3.17a)$$

$$\mathbf{W}_{db} \Delta B + \mathbf{W}_{dd} \Delta D + \mathbf{A}_d^T \Delta \Lambda = \mathbb{I}_{n_d} \quad (3.17b)$$

$$\mathbf{A}_b \Delta B + \mathbf{A}_d \Delta D = 0. \quad (3.17c)$$

Eliminating ΔB from (3.17c) and plugging in (3.17a) leads to,

$$\Delta \Lambda = \mathbf{A}_b^{-T} (\mathbf{W}_{bb} \mathbf{A}_b^{-1} \mathbf{A}_d - \mathbf{W}_{bd}) \Delta D \quad (3.18)$$

substituting into (3.17b),

$$\left(\mathbf{W}_{dd} + \mathbf{A}_d^T \mathbf{A}_b^{-T} \mathbf{W}_{bb} \mathbf{A}_b^{-1} \mathbf{A}_d - \mathbf{A}_d^T \mathbf{A}_b^{-T} \mathbf{W}_{bd} - \mathbf{W}_{db} \mathbf{A}_b^{-1} \mathbf{A}_d \right) \Delta D = \mathbb{I}_{n_d}. \quad (3.19)$$

Solving for ΔD leads to $\Delta D = (Z^T \mathbf{W} Z)^{-1} \cdot \square$

The above strategy is only valid when there are no active bounds at the solution. With this, the inverse of the reduced barrier Hessian matches the inverse of the reduced Hessian.

Moreover, if inertia correction is required at the solution ($\delta_c > 0$ or $\delta_d > 0$), the extracted matrix is the inverse of the *regularized* reduced barrier Hessian which is an artificial matrix. Although the regularization is an indication of a rank-deficient Jacobian or the presence of directions of negative curvature, no further information of the nature of the optimal point can be inferred from the regularized KKT matrix.

Finally, note that IPOPT does not make an explicit distinction between the variables x for solving the NLP. For the implementation of this strategy, we provide IPOPT with *pointers* that specify which variables x should be used as d to compute the reduced Hessian information. This can be done through modeling platforms such as AMPL.

The availability of reduced Hessian information can be valuable in the solution of large-scale optimal control problems and parameter and state estimation problems. For instance, the reduced Hessian matrix can be used for the analysis of singular optimal control problems and to infer controllability properties of complex dynamic systems [66]. In Chapters 4 and 6 we will also see that reduced Hessian information can also be used to perform inference analyzes of solutions of estimation problems.

Illustrative Example

Consider a simple NLP problem,

$$\min \quad (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 \tag{3.20a}$$

$$\text{s.t.} \quad x_1 + 2x_2 + 3x_3 = 0. \tag{3.20b}$$

The Lagrange function associated with this problem is,

$$\mathcal{L} = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + \lambda(x_1 + 2x_2 + 3x_3). \tag{3.21}$$

The first-order optimality conditions,

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial x_1} &= 2(x_1 - 1) + \lambda = 0 \\
 \frac{\partial \mathcal{L}}{\partial x_2} &= 2(x_2 - 2) + 2\lambda = 0 \\
 \frac{\partial \mathcal{L}}{\partial x_3} &= 2(x_3 - 3) + 3\lambda = 0 \\
 \frac{\partial \mathcal{L}}{\partial \lambda} &= x_1 + 2x_2 + 3x_3 = 0
 \end{aligned} \tag{3.22}$$

linearizing,

$$\begin{bmatrix} 2 & & 1 \\ & 2 & 2 \\ & & 2 & 3 \\ 1 & 2 & 3 & \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x_1} \\ \frac{\partial \mathcal{L}}{\partial x_2} \\ \frac{\partial \mathcal{L}}{\partial x_3} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{bmatrix} \tag{3.23}$$

from here, we can identify the structure of the KKT system,

$$\begin{bmatrix} \mathbf{H} & \mathbf{A}^T \\ \mathbf{A} & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{bmatrix}. \tag{3.24}$$

Here, the Hessian of the Lagrangian is equal to the barrier Hessian since we have no bounds.

We define the reduced Hessian matrix as,

$$\bar{H} = Z^T \mathbf{H} Z. \tag{3.25}$$

If we choose $b = x_1$ and $d = [x_2, x_3]$ we obtain,

$$Z = \begin{bmatrix} -(1)^{-1} [2 \ 3] \\ \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -2 & -3 \\ 1 & \\ & 1 \end{bmatrix} \tag{3.26}$$

the reduced Hessian is,

$$\bar{H} = Z^T \mathbf{H} Z = \begin{bmatrix} -2 & 1 & \\ -3 & & 1 \end{bmatrix} \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix} \begin{bmatrix} -2 & -3 \\ 1 & \\ & 1 \end{bmatrix} = \begin{bmatrix} 10 & 12 \\ 12 & 20 \end{bmatrix} \tag{3.27}$$

and the inverse of the reduced Hessian,

$$\bar{H}^{-1} = \begin{bmatrix} \bar{H}_{x_2, x_2}^{-1} & \bar{H}_{x_2, x_3}^{-1} \\ \bar{H}_{x_3, x_2}^{-1} & \bar{H}_{x_3, x_3}^{-1} \end{bmatrix} = \begin{bmatrix} 0.3571 & -0.2143 \\ -0.2143 & 0.1786 \end{bmatrix} \tag{3.28}$$

The procedure to extract reduced Hessian information from the primal-dual matrix can be explained by looking at the primal-dual system of our example (3.23). If we recursively place unit vectors in the right-hand side of the primal-dual system we can extract the inverse of the reduced Hessian. For instance,

$$\begin{bmatrix} 2 & & 1 \\ & 2 & 2 \\ & & 2 & 3 \\ 1 & 2 & 3 & \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \Delta x_2 = \bar{H}_{x_2, x_2}^{-1}, \Delta x_3 = \bar{H}_{x_2, x_3}^{-1} \quad (3.29)$$

and,

$$\begin{bmatrix} 2 & & 1 \\ & 2 & 2 \\ & & 2 & 3 \\ 1 & 2 & 3 & \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \Delta x_2 = \bar{H}_{x_3, x_2}^{-1}, \Delta x_3 = \bar{H}_{x_3, x_3}^{-1} \quad (3.30)$$

With this, we are implicitly forming the columns of the inverse of the reduced Hessian.

3.2.2 Nonlinear Programming Sensitivity

Note that problem (3.1) can be treated as a parametric programming problem in the data $\eta \in \mathcal{R}^{n_\eta}$. With this, the optimal values of the primal and dual variables can be treated as implicit functions of η . For a *sufficiently small* μ [41, 131], the KKT conditions of the barrier problem (3.3) can be expressed as,

$$\varphi(s_*(\eta), \eta) = 0. \quad (3.31)$$

Here, we are interested in computing fast approximate solutions for neighboring problems around an already available nominal solution $s_*(\eta_0)$. In order to do this, we make use of the following classical results.

Theorem 3.2.2. (*NLP Sensitivity*) [41, 25]. If $f(\cdot)$ and $c(\cdot)$ of the parametric problem $\mathcal{N}(\eta)$ are twice continuously differentiable in a neighborhood of the nominal solution $s_*(\eta_0)$ and this solution satisfies LICQ and SSOC, then

- $s_*(\eta_0)$ is an isolated local minimizer of $\mathcal{N}(\eta_0)$ and the associated Lagrange multipliers are unique.
- For η in a neighborhood of η_0 there exists a unique, continuous and differentiable vector function $s_*(\eta)$ which is a local minimizer satisfying LICQ and SSOC for $\mathcal{N}(\eta)$.
- There exists a positive Lipschitz constant L such that $\|s_*(\eta) - s_*(\eta_0)\| \leq L\|\eta - \eta_0\|$.
- There exists a positive Lipschitz constant L_f such that the optimal objective function values $f(s_*(\eta), \eta)$ and $f(s_*(\eta_0), \eta_0)$ satisfy $\|f(s_*(\eta), \eta) - f(s_*(\eta_0), \eta_0)\| \leq L_f\|\eta - \eta_0\|$.

An immediate consequence of these results is that we can apply the implicit function theorem to (3.31) at $s_*(\eta_0)$ to yield,

$$\bar{\mathbf{K}}_*(\eta_0) \frac{\partial s_*}{\partial \eta} = - \frac{\partial \varphi(s_*(\eta_0), \eta_0)}{\partial \eta} \quad (3.32)$$

where $\bar{\mathbf{K}}_*(\eta_0)$ is the primal-dual matrix of $\mathcal{N}(\eta_0)$ evaluated at $s_*(\eta_0)$. As sketched in Figure 3.2, first-order estimates of the solutions of neighboring problems can be obtained from the explicit form,

$$\tilde{s}(\eta) = s_*(\eta_0) + \frac{\partial s_*}{\partial \eta}(\eta - \eta_0) \quad (3.33)$$

where $\tilde{s}(\eta)$ is an approximate solution of $s_*(\eta)$. From continuity and differentiability of the optimal solution vector, there exists a positive Lipschitz constant L_s such that [33],

$$\|\tilde{s}(\eta) - s_*(\eta)\| \leq L_s \|\eta - \eta_0\|^2. \quad (3.34)$$

If the nominal solution satisfies LICQ and SSOC (e.g. no inertia correction required by IPOPT), then the primal-dual matrix is non-singular and can be reused to compute the sensitivity matrix from (3.32). Moreover, since the factorization of the primal-dual matrix³ $\bar{\mathbf{K}}_*(\eta_0)$ is already available from the solution of the nominal problem $\mathcal{N}(\eta_0)$, the

³If the primal-dual matrix $\bar{\mathbf{K}}_*$ is seen as a black box, this is equivalent to say that the augmented matrix in (3.6) is already factorized.

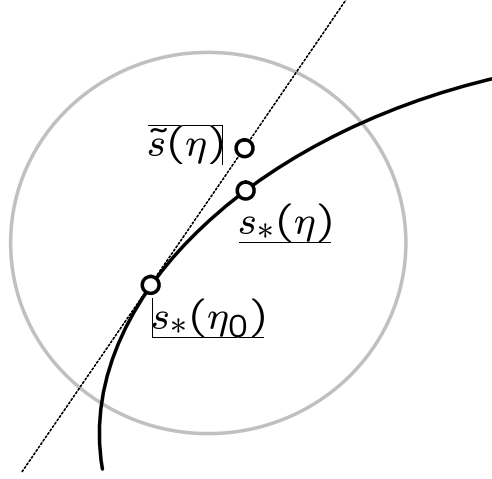


Figure 3.2: NLP sensitivity approximation.

computation of the sensitivity matrix from (3.32) requires n_η backsolves on the system,

$$\bar{\mathbf{K}}_*(\eta_0) \frac{\partial s_*}{\partial \eta}(:, j) = -\frac{\partial \varphi(s_*(\eta_0), \eta_0)}{\partial \eta}(:, j). \quad (3.35)$$

The step $\Delta s(\eta) = \tilde{s}(\eta) - s_*(\eta_0)$ in (3.33) can also be found by linearization of the KKT conditions (3.31) around $s_*(\eta_0)$ to give,

$$\begin{aligned} \bar{\mathbf{K}}_*(\eta_0) \Delta s(\eta) &= -(\varphi(s_*(\eta_0), \eta) - \varphi(s_*(\eta_0), \eta_0)) \\ &= -\varphi(s_*(\eta_0), \eta). \end{aligned} \quad (3.36)$$

Here, $\Delta s(\eta) = \tilde{s}(\eta) - s_*(\eta_0)$ is a Newton step taken from $s_*(\eta_0)$ towards the solution of a neighboring problem $\mathcal{N}(\eta)$ so that $\tilde{s}(\eta)$ satisfies (3.33)-(3.34). Computing this step requires a *single backsolve* which can be performed orders of magnitude faster than the factorization of the KKT matrix. Consequently, this is a fast way of computing approximate solutions to neighboring problems.

Since the approximate solution $\tilde{s}(\eta)$ is accurate to first order, we can use it as the initial guess $s_o(\eta)$ to warm-start the neighboring NLP $\mathcal{N}(\eta)$. Furthermore, we can reuse the KKT matrix $\bar{\mathbf{K}}_*(\eta_0)$ to perform fast fixed-point iterations on the system,

$$\bar{\mathbf{K}}_*(\eta_0) \Delta s_i = -\varphi(s_i(\eta), \eta) \quad (3.37)$$

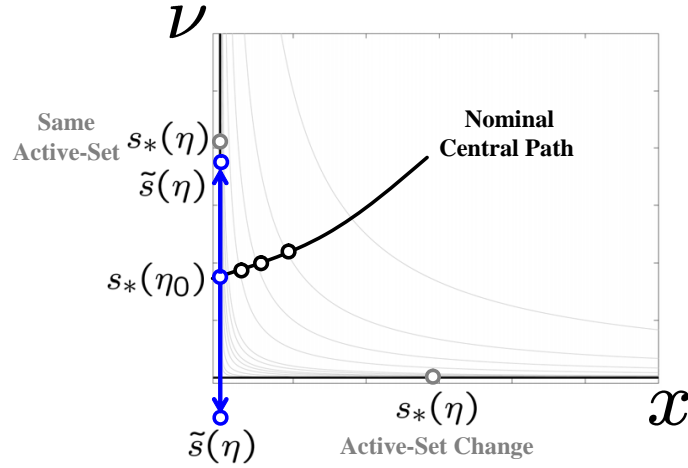


Figure 3.3: NLP sensitivity in the face of active-set changes.

with $\Delta s_i = s_{i+1}(\eta) - s_i(\eta)$ and $s_o(\eta) = s_*(\eta_0)$. With this, we can refine the warm-start point of the perturbed problem $\mathcal{N}(\eta)$ until no further progress can be made with the fixed KKT matrix. For sufficiently small perturbations, these fast fixed-point iterations can converge to the solution of the perturbed problem $s_*(\eta)$ [33]. However, for large perturbations, the KKT matrix needs to be reevaluated and refactorized (i.e. the NLP may need to be solved again).

The implementation of NLP sensitivity capabilities in IPOPT is straightforward. We treat the nominal data vector η_0 as variables that are fixed during the solution of the NLP. Once the optimal solution is found, we substitute the perturbed values η on the right-hand side of the primal-dual system (A.9) to obtain the approximate solution. This can be done recursively for fixed-point iterations.

Note that continuity of the solution path $s_*(\eta)$ around the nominal solution $s_*(\eta_0)$ implies that the approximation (3.33) is only valid in a neighborhood where the active-set of $s_*(\eta_0)$ remains unchanged [41]. In the presence of an active-set change, the linearization of the complementarity relaxation in (3.4) (with small μ) will tend to drive the perturbed Newton step (3.36) outside of the feasible region. This scenario is illustrated in Figure 3.3.

These observations are of practical relevance in the context of warm-starting strategies for interior-point solvers. Imagine that we warm-start the solution of problem $\mathcal{N}(\eta)$ with the nominal solution $s_*(\eta_0)$ and we set μ to a small value. If the solution $s_*(\eta)$ has the same active-set, the first Newton iteration taken by the interior-point solver will take us close to the optimal solution. However, when the solution $s_*(\eta)$ has a different active-set, the first Newton step will take us outside of the feasible region. The solver will cut back the step and will tend to stay close to the boundary making slow progress towards the solution. Note that even if we know the right active-set of the solution, guesses of the values of the bound multipliers will be needed as well but these are difficult to obtain. This is an important disadvantage of interior-point solvers when compared to active-set solvers. This is particularly relevant in applications that rely on subsequent solutions of similar NLPs (e.g. on-line applications or mixed-integer programming algorithms). In order to account for active-set changes, it is possible to compute Newton steps from the solution of multiple barrier problems along the central path. This strategy has been previously proposed to warm-start linear and quadratic programming problems [38, 50]. A problem that arises in infeasible path solvers like IPOPT is that, in most cases, the solver does not stay close to the central path so it is difficult to keep track of solutions of barrier problems. In addition, storing multiple KKT matrices is impractical in large-scale applications. In Section 3.2.3 we will present a practical strategy to overcome some of these limitations.

Illustrative Example

Consider the parametric NLP problem presented by Ganesh and Biegler [49],

$$\begin{aligned}
 \min \quad & x_1^2 + x_2^2 + x_3^2 \\
 \text{s.t.} \quad & 6x_1 + 3x_2 + 2x_3 - \eta_1 = 0 \\
 & \eta_2 x_1 + x_2 - x_3 - 1 = 0 \\
 & x_1, x_2, x_3 \geq 0.
 \end{aligned} \tag{3.38}$$

The problem data is given by $\eta = [\eta_1, \eta_2]$. The Lagrange function is given by,

$$\begin{aligned} \mathcal{L} = & x_1^2 + x_2^2 + x_3^2 + \lambda_1(6x_1 + 3x_2 + 2x_3 - \eta_1) + \lambda_2(\eta_2 x_1 + x_2 - x_3 - 1) \\ & - \nu_{x_1} x_1 - \nu_{y_1} x_2 - \nu_{x_3} x_3 \end{aligned} \quad (3.39)$$

Using a primal-dual barrier approach, the KKT conditions of the barrier problem become:

$$\begin{aligned} 2x_1 + 6\lambda_1 + \eta_2\lambda_2 - \nu_{x_1} &= 0 \\ 2x_2 + 3\lambda_1 + \lambda_2 - \nu_{y_1} &= 0 \\ 2x_3 + 2\lambda_1 - \lambda_2 - \nu_{y_2} &= 0 \\ 6x_1 + 3x_2 + 2x_3 - \eta_1 &= 0 \\ \eta_2 x_1 + x_2 - x_3 - 1 &= 0 \\ \nu_{x_1} x_1 - \mu &= 0 \\ \nu_{x_2} x_2 - \mu &= 0 \\ \nu_{x_3} x_3 - \mu &= 0 \\ x_1, x_2, x_3, \nu_{x_1}, \nu_{x_2}, \nu_{x_3} &\geq 0. \end{aligned} \quad (3.40)$$

Linearizing around an arbitrary point s^k we obtain the primal-dual system,

$$\begin{bmatrix} 2 & & & 6 & \eta_2 & -1 & & & \\ & 2 & & 3 & 1 & & -1 & & \\ & & 2 & 2 & -1 & & & -1 & \\ 6 & 3 & 2 & & & & & & \\ \eta_2 & 1 & -1 & & & & & & \\ v_{x_1}^k & & & & x_1^k & & & & \\ & v_{x_2}^k & & & & x_2^k & & & \\ & & v_{x_3}^k & & & & x_3^k & & \end{bmatrix} \begin{bmatrix} \Delta x_1^k \\ \Delta x_2^k \\ \Delta x_3^k \\ \Delta \lambda_1^k \\ \Delta \lambda_2^k \\ \Delta \nu_{x_1}^k \\ \Delta \nu_{x_2}^k \\ \Delta \nu_{x_3}^k \end{bmatrix} = - \begin{bmatrix} 2x_1^k + 6\lambda_1^k + \eta_2\lambda_2^k - \nu_{x_1}^k \\ 2x_2^k + 3\lambda_1^k + \lambda_2^k - \nu_{x_2}^k \\ 2x_3^k + 2\lambda_1^k - \lambda_2^k - \nu_{x_3}^k \\ 6x_1^k + 3x_2^k + 2x_3^k - \eta_1 \\ \eta_2 x_1^k + x_2^k - x_3^k - 1 \\ \nu_{x_1}^k x_1^k - \mu \\ \nu_{x_2}^k x_2^k - \mu \\ \nu_{x_3}^k x_3^k - \mu \end{bmatrix} \quad (3.41)$$

From Table 3.1 we see that the solution of the NLP for the nominal value of $\eta_0 = [\eta_{1,0} \ \eta_{2,0}] = [4.5 \ 1]$ is $s^*(\eta_0) = [0.5 \ 0.5 \ 0 \ 0 \ -1 \ 0 \ 0 \ 1]$. At this point, the primal-dual system becomes,

$$\begin{bmatrix}
2 & & & 6 & \eta_{2,0} & -1 & & & \\
& 2 & & 3 & 1 & & -1 & & \\
& & 2 & 2 & -1 & & & -1 & \\
6 & 3 & 2 & & & & & & \\
\eta_{2,0} & 1 & -1 & & & & & & \\
O(\mu) & & & & 0.5 & & & & \\
& O(\mu) & & & & 0.5 & & & \\
& & 1 & & & & O(\mu) & &
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2 \\
\Delta x_3 \\
\Delta \lambda_1 \\
\Delta \lambda_2 \\
\Delta \nu_{x_1} \\
\Delta \nu_{x_2} \\
\Delta \nu_{x_3}
\end{bmatrix}
= -
\begin{bmatrix}
2x_1^* + 6\lambda_1^* + \eta_{2,0}\lambda_2^* - \nu_{x_1}^* \\
2x_2^* + 3\lambda_1^* + \lambda_2^* - \nu_{x_2}^* \\
2x_3^* + 2\lambda_1^* - \lambda_2^* - \nu_{x_3}^* \\
6x_1^* + 3x_2^* + 2x_3^* - \eta_{1,0} \\
\eta_{2,0}x_1^* + x_2^* - x_3^* - 1 \\
\nu_{x_1}^*x_1^* - \mu \\
\nu_{x_2}^*x_2^* - \mu \\
\nu_{x_3}^*x_3^* - \mu
\end{bmatrix}
\quad (3.42)$$

where μ is a small value. Assume that we want to find the solution of the perturbed problem with $\eta = [4 \ 1]$. The exact solution of this problem is given in Table 3.1. Note that the active-set remains unchanged for this perturbation. If we replace η_0 with η in the right-hand side of (3.42), the computed step returns $\tilde{s}(\eta) = [0.333 \ 0.666 \ 0 \ 0.2222 \ -2 \ 0 \ 0 \ 2.44]$. This is a perfect approximation of the solution $s^*(\eta) = [0.333 \ 0.666 \ 0 \ 0.2222 \ -2 \ 0 \ 0 \ 2.44]$. Now consider the perturbation with $\eta = [5 \ 1]$. From the table we see that x_3 becomes inactive (i.e. there is a change in the active-set). The sensitivity solution is $\tilde{s}(\eta) = [0.666 \ 0.333 \ 0 \ -0.2222 \ 0 \ 0 \ 0 \ -0.444]$ which differs from the optimal solution $s^*(\eta) = [0.6327 \ 0.388 \ 0.0204 \ -0.163 \ -0.286 \ 0 \ 0 \ 0]$. In particular, note that the predicted bound multiplier ν_{x_3} is negative (outside of the feasible region). This illustrates the limitations of NLP sensitivity in handling active-set changes.

η_1	η_2	x_1	x_2	x_3	λ_1	λ_2	ν_{x_1}	ν_{x_2}	ν_{x_3}
4.0	1	0.333	0.666	0	0.222	-2	0	0	2.444
4.5	1	0.5	0.5	0	0	-1	0	0	1
5	1	0.6327	0.388	0.0204	-0.163	-0.286	0	0	0

Table 3.1: Solutions of parametric NLP example.

3.2.3 Fix-Relax Strategy

NLP sensitivity allows to analyze the effect of perturbations on the problem data around the nominal value η_0 which is known *a priori*. Consider now the scenario in which we

would like to study the effect of perturbations on the values of *internal variables* around a given solution. In other words, imagine that the data η_0 is not known *a priori* but is to be determined from the solution of the NLP. As we will see later, this capability has many applications. For instance, it can be used to compute fast NLP sensitivity approximations in the face of active-set changes.

We consider the variable partition $x^T = [g^T \ u^T]$ where $g \in \mathbb{R}^{n_g}$ is a vector of *fixed* variables. We study the effect of perturbations on the *unfixed* variables $u \in \mathbb{R}^{n_u}$ where $n_u = n_x - n_g$. For this, we define the nominal value u_0 obtained from the solution of a nominal NLP and the perturbed value u_p .

According to the variable partition, we can represent the primal-dual system (3.4) as,

$$\begin{bmatrix} \mathbf{H}_{gg} & \mathbf{H}_{gu} & \mathbf{A}_g^T & -\mathbb{I}_{n_g} & 0 \\ \mathbf{H}_{ug} & \mathbf{H}_{uu} & \mathbf{A}_u^T & 0 & -\mathbb{I}_{n_u} \\ \mathbf{A}_g & \mathbf{A}_u & 0 & 0 & 0 \\ \mathbf{V}_g & 0 & 0 & \mathbf{G} & 0 \\ 0 & \mathbf{V}_u & 0 & 0 & \mathbf{U} \end{bmatrix} \begin{bmatrix} \Delta g \\ \Delta u \\ \Delta \lambda \\ \Delta \nu_g \\ \Delta \nu_u \end{bmatrix} = - \begin{bmatrix} r_g \\ r_u \\ r_\lambda \\ r_{\nu_g} \\ r_{\nu_u} \end{bmatrix}. \quad (3.43)$$

In compact form,

$$\bar{\mathbf{K}}_* \Delta s = -r_s \quad (3.44)$$

where $\nabla_s \mathcal{L} = 0$ at the current solution. The primal-dual matrix on the left-hand side is already factorized at the solution. To perturb the internal variable u , we impose the constraint,

$$\Delta u = -(u_0 - u_p). \quad (3.45)$$

By doing so, the optimality conditions of u do not hold anymore and need to be discarded. We want to avoid dropping the second row in (3.43) since this would change the structure of the KKT matrix and we will need to refactorize it. In order to avoid this, we eliminate

these rows by adding slack variables $\Delta\bar{\nu}_u$. With this, system (3.43) becomes,

$$\left[\begin{array}{ccccc|c} \mathbf{H}_{gg} & \mathbf{H}_{gu} & \mathbf{A}_g^T & -\mathbb{I}_{n_g} & 0 & 0 \\ \mathbf{H}_{ug} & \mathbf{H}_{uu} & \mathbf{A}_u^T & 0 & -\mathbb{I}_{n_u} & \mathbb{I}_{n_u} \\ \mathbf{A}_g & \mathbf{A}_u & 0 & 0 & 0 & 0 \\ \mathbf{V}_g & 0 & 0 & \mathbf{G} & 0 & 0 \\ 0 & \mathbf{V}_u & 0 & 0 & \mathbf{U} & 0 \\ \hline 0 & \mathbb{I}_{n_u} & 0 & 0 & 0 & 0 \end{array} \right] \left[\begin{array}{c} \Delta g \\ \Delta u \\ \Delta \lambda \\ \Delta \nu_g \\ \Delta \nu_u \\ \Delta \bar{\nu}_u \end{array} \right] = - \left[\begin{array}{c} r_g \\ r_u \\ r_\lambda \\ r_{\nu_g} \\ r_{\nu_u} \\ u_0 - u_p \end{array} \right]. \quad (3.46)$$

if we define $\mathbf{E} = [0 \ \mathbb{I}_{n_u} \ 0 \ 0 \ 0 \ 0]$, we can represent the above linear system as,

$$\left[\begin{array}{c|c} \bar{\mathbf{K}}_* & \mathbf{E}^T \\ \hline \mathbf{E} & 0 \end{array} \right] \left[\begin{array}{c} \Delta s \\ \Delta \bar{\nu}_u \end{array} \right] = - \left[\begin{array}{c} r_s \\ u_0 - u_p \end{array} \right] \quad (3.47)$$

where $\Delta s = \tilde{s}(u_p) - s_*(u_0)$. In other words, the internal variable u can be seen as the problem data η . Furthermore, $\tilde{s}(u_p)$ is an approximation of the optimal solution $s_*(u_p)$. In addition, the slack variables $\Delta\bar{\nu}_u$ can be seen as the multipliers of the artificial constraint (3.45). To compute this approximate solution efficiently, we reuse the factorization of $\bar{\mathbf{K}}_*$ following a Schur decomposition,

$$\mathbf{S}\Delta\bar{\nu}_u = \mathbf{E}\bar{\mathbf{K}}_*^{-1}r_s - (u_0 - u_p) \quad (3.48a)$$

$$\bar{\mathbf{K}}_*\Delta s = -(r_s + \mathbf{E}^T\Delta\bar{\nu}_u). \quad (3.48b)$$

The Schur complement matrix $\mathbf{S} \in \Re^{n_u \times n_u}$ is given by,

$$\mathbf{S} = -(\mathbf{E}\bar{\mathbf{K}}_*^{-1}\mathbf{E}^T). \quad (3.49)$$

The product matrix $\mathbf{P} = \bar{\mathbf{K}}_*^{-1}\mathbf{E}^T$ is computed from backsolves on the system,

$$\bar{\mathbf{K}}_*\mathbf{P}(:,j) = \mathbf{E}^T(:,j). \quad (3.50)$$

Therefore, forming the Schur complement requires n_u backsolves. The step (3.48a) requires a dense factorization of \mathbf{S} . To refine the approximate solutions, we can perform fixed point

iterations on the system,

$$\left[\begin{array}{c|c} \bar{\mathbf{K}}_* & \mathbf{E}^T \\ \hline \mathbf{E} & 0 \end{array} \right] \left[\begin{array}{c} \Delta s_i \\ \Delta \bar{\nu}_{u,i} \end{array} \right] = - \left[\begin{array}{c} r_s(s_i(u_p)) \\ u_0 - u_p \end{array} \right] \quad (3.51)$$

with $s_{i+1}(u_p) = s_i(u_p) + \Delta s_i$ and $s_0(u_p) = s_*(u_0)$.

The above procedure can be used to compute approximate solutions for perturbations on internal variables. Perturbations can also be performed on the dual variables ν (i.e. active constraints at the solution can also be made inactive). Therefore, this fix-relax strategy can be used as a Schur update strategy used in some active-set QP solvers [8]. Since this strategy provides guesses for both the primal and dual variables, it can be used to warm-start interior-point methods in the face of active-set changes.

Illustrative Example

We revisit the active-set change scenario of problem (3.38). As we have seen, the perturbation $\eta = [5 \ 1]$ generates the inconsistent approximation,

$$\tilde{s}(\eta) = [0.666 \ 0.333 \ 0 \ -0.2222 \ 0 \ 0 \ 0 \ -0.444].$$

Following an active-set strategy, we fix the negative bound multiplier to zero. This is done by adding the constraint $\Delta \nu_{x_3} = -(1 - 0)$ where 1 is the optimal value of ν_{x_3} at the nominal solution ($\eta_0 = [4.5 \ 1]$). By relaxing the complementarity relaxation in (3.42) through the slack variable $\bar{\nu}_{x_3}$, we obtain,

$$\left[\begin{array}{cccccc|c} 2 & & 6 & \eta_{2,0} & -1 & & \\ & 2 & & 3 & 1 & -1 & \\ & & 2 & 2 & -1 & & -1 \\ 6 & 3 & 2 & & & & \\ \eta_{2,0} & 1 & -1 & & & & \\ O(\mu) & & & & 0.5 & & \\ & O(\mu) & & & & 0.5 & \\ & & 1 & & & O(\mu) & 1 \\ \hline & & & & & 1 & \end{array} \right] \left[\begin{array}{c} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda_1 \\ \Delta \lambda_2 \\ \Delta \nu_{x_1} \\ \Delta \nu_{x_2} \\ \Delta \nu_{x_3} \\ \Delta \bar{\nu}_{x_3} \end{array} \right] = - \left[\begin{array}{c} 2x_1^* + 6\lambda_1^* + \eta_{2,0}\lambda_2^* - \nu_{x_1}^* \\ 2x_2^* + 3\lambda_1^* + \lambda_2^* - \nu_{x_2}^* \\ 2x_3^* + 2\lambda_1^* - \lambda_2^* - \nu_{x_3}^* \\ 6x_1^* + 3x_2^* + 2x_3^* - \eta_{1,0} \\ \eta_{2,0}x_1^* + x_2^* - x_3^* - 1 \\ \nu_{x_1}^* x_1^* - \mu \\ \nu_{x_2}^* x_2^* - \mu \\ \nu_{x_3}^* x_3^* - \mu \\ \hline 1 - 0 \end{array} \right] \quad (3.52)$$

The solution of this linear system (e.g. through a Schur decomposition) leads to the approximate solution $\tilde{s}(\eta) = [0.6327 \ 0.388 \ 0.0204 \ -0.163 \ -0.286 \ 0 \ 0 \ 0]$ which is a perfect approximation of the optimal solution $s^*(\eta) = [0.6327 \ 0.388 \ 0.0204 \ -0.163 \ -0.286 \ 0 \ 0 \ 0]$. As can be seen, the fix-relax strategy is equivalent to solve an active-set QP using Schur complement techniques [9].

3.3 Concluding Remarks

In this chapter, we have contrasted advantages and disadvantages of full-space interior-point solvers in the context of DAE-constrained optimization. In particular, we have seen that the ability to exploit the sparsity of the discretized DAE model directly from the Karush-Kuhn-Tucker (KKT) matrix leads to a highly favorable computational complexity.

We have presented practical strategies to verify the satisfaction of sufficient second order conditions directly from the KKT matrix of the interior-point solver. Here, we make use of basic inertia results and Debreu's lemma to prove that, if the KKT matrix has the right inertia at the solution, the solution is an optimal point satisfying sufficient second order conditions.

We also demonstrate that it is possible to re-use the factorization of the full-space KKT matrix available at the solution to extract reduced Hessian information through inexpensive backsolves. In addition, we explain how to re-use the factorization to compute fast approximate solutions for parametric NLPs using NLP sensitivity concepts. We demonstrate that these inexpensive sensitivity approximations are accurate and can be used to warm-start interior-point solvers. In the following sections, we will describe how to use these capabilities in the context of model-based operations.

Part II

Model-Based Operations

Chapter 4

Parameter Estimation

Having established the basic components of the computational framework, we now use these developments to solve different optimization problems arising in the development and implementation of model-based applications. In this chapter, we discuss the solution of large-scale parameter estimation problems.

4.1 Introduction

Mathematical modeling of industrial processes is a fundamental but difficult task that motivates a huge amount of research effort. The development of these detailed models follows a number of steps including repetitive model discrimination and experimental design steps under controlled laboratory and pilot plant conditions. Once the initial model structure is developed, the model undergoes a parameter tuning stage using on-line data extracted from the full-scale process. This stage is complicated since the data tends to be noisy and since there exist uncertain phenomena in the process that cannot be captured by the model. The solution of parameter estimation problems is a key repetitive task during both the development and implementation of the model.

Well known parameter estimation theory and methods have been applied only recently to large-scale rigorous process models [40, 116, 14]. In many cases, the model is so complex and the amounts of data are so large that the estimation problem has to be simplified (e.g. trial and error). Unfortunately, these nonsystematic approaches can become time-consuming and may yield parameters with large confidence regions and lack of robustness in the predictions.

Here, we are interested in exploiting the advantages of the proposed computational strategies to solve estimation problems incorporating large-scale process models and multiple data sets.

4.2 Maximum-Likelihood Formulations

Assume that we have N_S measurement data sets reflecting the operating conditions of a given process under different scenarios. Each data set k contains input measurements u_k^M and measurements of the associated output responses y_k^M . Using this information, we seek to estimate the parameters p_k and Π of a given DAE model. Here, p_k are scenario-dependent parameters. These can be used to account for uncertain phenomena in the process such as catalyst deactivation, coking, fouling, etc. Parameters Π are scenario-independent parameters. These can be used to account for fundamental parameters such as Arrhenius terms that remain constant under different operating conditions. To estimate these parameters, we can use the standard least-squares formulation,

$$\begin{aligned}
\min_{\Pi, p_k} \quad & \Phi := \frac{1}{2} \sum_{k=1}^{N_S} (y_k - y_k^M)^T \mathbf{V}_y^{-1} (y_k - y_k^M) \\
s.t. \quad & \\
& \frac{dz_k^d(t)}{dt} = \mathbf{f}_k^d(z_k^d(t), z_k^a(t), u_k^M, p_k, \Pi) \\
& 0 = \mathbf{f}_k^a(z_k^d(t), z_k^a(t), u_k^M, p_k, \Pi) \\
& y_k = \chi_k(z_k^d(t), z_k^a(t), u_k^M, p_k, \Pi) \\
& 0 \leq \mathbf{h}_k(z_k^d(t), z_k^a(t), u_k^M, p_k, \Pi) \\
& z_k(0) = z_{0,k}, \quad k = 1, \dots, N_S
\end{aligned} \tag{4.1}$$

where $\mathbf{f}_k^d(\cdot), \mathbf{f}_k^a(\cdot)$ are differential and algebraic equations, respectively, defined for each data set $k = 1, \dots, N_S$. Symbol $z_k^d(t)$ denotes the differential state variables with initial conditions $z_{0,k}$, $z_k^a(t)$ are algebraic state variables and the independent variable t is either temporal or spatial. Equations $\chi_k(\cdot)$ are used to map the model variables to the measured outputs y_k . The equations $\mathbf{h}_k(\cdot)$ are general inequality constraints. The matrix \mathbf{V}_y is an

approximation of the covariance matrix of the output measurements.

In standard least-squares formulation the inputs u_k are fixed to the measurements u_k^M . Since in many cases the inputs contain measurement errors, this approach is well-known to give *biased* parameters [89, 69, 7]. In order to avoid this, we consider an errors-in-variables-measured (EVM) formulation. This formulation accounts for errors in all the measured variables (input and outputs). Because of this, the EVM approach can be seen as a simultaneous estimation/data reconciliation strategy that can be used to find more reliable parameters [82, 71, 2]. The general EVM formulation resembles that of the standard least-squares (4.1) but, in this case, the inputs in every data set k become decision variables. Upon addition of terms in the objective function that account for allowed adjustments of the input variables, the estimation problem becomes,

$$\begin{aligned}
\min_{\Pi, p_k, u_k} \quad & \Phi := \frac{1}{2} \sum_{k=1}^{N_S} (y_k - y_k^M)^T \mathbf{V}_y^{-1} (y_k - y_k^M) + (u_k - u_k^M)^T \mathbf{V}_u^{-1} (u_k - u_k^M) \\
s.t. \quad & \\
& \frac{dz_k^d(t)}{dt} = \mathbf{f}_k^d(z_k^d(t), z_k^a(t), u_k, p_k, \Pi) \\
& 0 = \mathbf{f}_k^a(z_k^d(t), z_k^a(t), u_k, p_k, \Pi) \\
& y_k = \chi_k(z_k^d(t), z_k^a(t), u_k, p_k, \Pi) \\
& 0 \leq \mathbf{h}_k(z_k^d(t), z_k^a(t), u_k, p_k, \Pi) \\
& z_k(0) = z_{0,k}, \quad k = 1, \dots, N_S.
\end{aligned} \tag{4.2}$$

Where \mathbf{V}_u is an approximation of the covariance matrix of the input measurements. The EVM approach corrects for measurement errors in all these variables and obtains less biased parameters. However, this comes at the expense of a significant increase in the number of degrees of freedom. Consequently, solutions of EVM problems are often considered to be computationally intensive.

Estimation problems are in general highly complex optimization problems. The first issue arises from the fact that the number of constraints increases with the number of data sets. Since in this context the constraints include a large and highly nonlinear DAE

model, the problem becomes computationally expensive as we add information into the problem. Another issue is that it is often difficult to determine the number of parameters that can be estimated reliably from the available measurement information. If the data are not informative enough, the problem can become ill-posed (e.g. different combinations of parameters give the same objective function value) and can become difficult to converge. Finally, EVM problems are especially difficult to solve because they involve a large number of degrees of freedom. For instance, the number of iterations required by a Quasi-Newton algorithm is directly related to the number of degrees of freedom [93].

4.3 Solution with IPOPT

We propose to use a full-discretization approach coupled to a full-space interior point solver to solve the estimation problems (4.1)-(4.2). As we have seen in Chapter 3, this approach presents a very favorable computational complexity. In addition, since it is possible to use cheap exact first and second order derivative information for IPOPT, the convergence properties of the algorithm are not affected by the number of degrees of freedom. Moreover, the solver is quite effective in handling negative curvature issues arising from ill-posedness and nonlinearity of the NLP.

On the other hand, we have also seen that when the KKT matrix (3.6) becomes very large (e.g. as we increase the number of data sets), its factorization can take an excessive amount of time or it may not even be possible due to computer memory bottlenecks. Another practical issue arising in this approach is that the estimation problems are solved as general NLPs. Consequently, we lose insight on the specifics of the application and it is thus difficult to interpret the nature of the solution. This is a particularly relevant issue in parameter estimation since it is often necessary to quantify the reliability of the parameter estimates (inference analysis).

In the following section, we discuss how to use the generalized multi-scenario decompo-

sition strategy proposed by Laird and Biegler [78, 132] to avoid memory bottlenecks in parameter estimation problems. We then discuss how to use reduced Hessian information extracted from IPOPT to perform inference analysis studies.

4.3.1 Exploitation of Multi-Set Structure

Note that the variables in (4.1)-(4.2) are entirely independent between scenarios or data sets and the only *complicating variables* are the parameters Π . If we define linking variables and equations to deal with Π , we can write problem (3.1) as a generalized multi-scenario problem of the form,

$$\begin{aligned} \min_{x_k, \Pi} \quad & \sum_{k=1}^{N_S} f_k(x_k) \\ \text{s.t.} \quad & \left. \begin{aligned} c_k(x_k) &= 0, \\ S_k x_k &\geq 0, \\ D_k x_k - \bar{D}_k \Pi &= 0 \end{aligned} \right\} k = 1, \dots, N_S \end{aligned} \quad (4.3)$$

where x_k contains all the parameters and variables corresponding to the discretization of the DAEs for a particular scenario k , Π is the linking variables vector, matrix D_k extracts the components corresponding to the parameters Π from the x_k vector and matrix \bar{D}_k assigns the extracted components ($D_k x_k$) to the linking variables vector Π . Matrix S_k extracts the components of x_k that are actually bounded such that, if all of the scenarios have the same structure, we can set $S_1 = \dots = S_{N_S}$. Otherwise, it is possible to incorporate scenarios with heterogeneous structures (e.g. different DAE models). Finally, note that the above formulation is considered for implementation purposes only since it allows to specify the multi-scenario problem through individual NLP instances.

Using a barrier formulation, this problem can be converted to:

$$\begin{aligned} \min_{x_k, \Pi} \quad & \sum_{k=1}^{N_S} \{f_k(x_k) - \mu \sum_j \ln[(S_k x_k)^{(j)}]\} \\ \text{s.t.} \quad & \left. \begin{aligned} c_k(x_k) &= 0, \\ D_k x_k - \bar{D}_k \Pi &= 0 \end{aligned} \right\} k = 1, \dots, N_S \end{aligned} \quad (4.4)$$

where indices j correspond to scalar elements of the vector $(S_k x_k)$. Defining the Lagrange function of the barrier problem (4.4),

$$\begin{aligned}\mathcal{L}(x, \lambda, \sigma, \Pi) &= \sum_{k=1}^{N_S} \bar{\mathcal{L}}_k(x_k, \lambda_k, \sigma_k, \Pi) \\ &= \sum_{k=1}^{N_S} \{ \mathcal{L}_k(x_k, \lambda_k, \sigma_k, \Pi) - \mu \sum_j \ln[(S_k x_k)^{(j)}] \} \end{aligned} \quad (4.5)$$

$$= \sum_{k=1}^{N_S} \{ f_k(x_k) - \mu \sum_j \ln[(S_k x_k)^{(j)}] + c_k(x_k)^T \lambda_k + [D_k x_k - \bar{D}_k \Pi]^T \sigma_k \} \quad (4.6)$$

with multipliers λ_k and σ_k . Defining $G_k = \text{diag}(S_k x_k)$ leads to the primal-dual form of the first-order optimality conditions, written as:

$$\left. \begin{aligned} \nabla_{x_k} f_k(x_k) + \nabla_{x_k} c_k(x_k) \lambda_k + D_k^T \sigma_k - S_k^T \nu_k &= 0 \\ c_k(x_k) &= 0 \\ D_k x_k - \bar{D}_k \Pi &= 0 \\ G_k \nu_k - \mu e &= 0 \end{aligned} \right\} \quad k = 1, \dots, N_S \quad (4.7)$$

$$- \sum_{k=1}^{N_S} \bar{D}_k^T \sigma_k = 0$$

where we define $e^T = [1, 1, \dots, 1]$. Writing the Newton step for (4.7) at iteration ℓ leads to:

$$\left. \begin{aligned} \nabla_{x_k x_k} \mathcal{L}_k^\ell \Delta x_k + \nabla_{x_k} c_k^\ell \Delta \lambda_k + D_k^T \Delta \sigma_k - S_k^T \Delta \nu_k &= -(\nabla_{x_k} \mathcal{L}_k^\ell - S_k^T \nu_k^\ell) \\ \nabla_{x_k} c_k^\ell \Delta x_k &= -c_k^\ell \\ D_k \Delta x_k - \bar{D}_k \Delta \Pi &= -D_k x_k^\ell + \bar{D}_k \Pi^\ell \\ V_k^\ell S_k \Delta x_k + G_k \Delta \nu_k &= \mu e - G_k \nu_k^\ell \end{aligned} \right\} \quad k = 1, \dots, N_S \quad (4.8)$$

$$- \sum_{k=1}^{N_S} \bar{D}_k^T \Delta \sigma_k = \sum_{k=1}^{N_S} \bar{D}_k^T \sigma_k^\ell$$

where the superscript ℓ indicates that the quantity is evaluated at the point $(x_k^\ell, \lambda_k^\ell, \sigma_k^\ell, \nu_k^\ell, \Pi^\ell)$.

Eliminating $\Delta\nu_k$ from the resulting linear equation gives the primal-dual *augmented system*

$$\left. \begin{aligned} H_k^\ell \Delta x_k + \nabla_{x_k} c_k^\ell \Delta \lambda_k + D_k^T \Delta \sigma_k &= -\nabla_{x_k} \bar{\mathcal{L}}_k^\ell \\ \nabla_{x_k} c_k^\ell \Delta x_k &= -c_k^\ell \\ D_k \Delta x_k - \bar{D}_k \Delta \Pi &= -D_k x_k^\ell + \bar{D}_k \Pi^\ell \\ -\sum_{k=1}^{N_S} \bar{D}_k^T \Delta \sigma_k &= \sum_{k=1}^{N_S} \bar{D}_k^T \sigma_k^\ell \end{aligned} \right\} \quad k = 1, \dots, N_S \quad (4.9)$$

where $H_k^\ell = \nabla_{x_k x_k} \mathcal{L}_k^\ell + S_k^T (G_k^\ell)^{-1} V_k^\ell S_k$, and $V_k = \text{diag}(\nu_k)$.

According to the IPOPT algorithm [121], the linear system (4.9) is modified as necessary by adding diagonal terms. Diagonal elements are added to the block Hessian terms in the *augmented system* to handle nonpositive curvature ($\delta_x \mathbb{I}$) and to the lower right corner in each block to handle temporary dependencies in the constraints ($\delta_c \mathbb{I}$). Applying these modifications, linear system (4.9) can be written with a block bordered diagonal (arrowhead) structure given by:

$$\begin{bmatrix} W_1 & & & & A_1 \\ & W_2 & & & A_2 \\ & & W_3 & & A_3 \\ & & & \ddots & \vdots \\ & & & & W_{N_S} & A_{N_S} \\ A_1^T & A_2^T & A_3^T & \cdots & A_{N_S}^T & \delta_x \mathbb{I} \end{bmatrix} \cdot \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \\ \vdots \\ \Delta v_{N_S} \\ \Delta \Pi \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{N_S} \\ r_\Pi \end{bmatrix} \quad (4.10)$$

where $r_k^T = -[(\nabla_{x_k} \mathcal{L}_k^\ell)^T, (c_k^\ell)^T, (D_k x_k^\ell - \bar{D}_k \Pi^\ell)^T]$, $\Delta v_k^T = [\Delta x_k^T, \Delta \lambda_k^T, \Delta \sigma_k^T]$, $A_k^T = [0 \ 0 \ -\bar{D}_k^T]$,

$$W_k = \begin{bmatrix} H_k^\ell + \delta_x \mathbb{I} & \nabla_{x_k} c_k^\ell & D_k^T \\ (\nabla_{x_k} c_k^\ell)^T & -\delta_c \mathbb{I} & 0 \\ D_k & 0 & -\delta_c \mathbb{I} \end{bmatrix}$$

for $k = 1, \dots, N_S$, and $r_\Pi = \sum_{k=1}^{N_S} \bar{D}_k^T \sigma_k^\ell$. The IPOPT algorithm requires the solution of the *augmented system* (4.10), at each iteration along with the determination of its inertia. This linear system can be solved, in principle, with any general direct linear solver configured

with IPOPT. However, as the problem size grows, the time and memory requirements can make this approach intractable. To avoid this, we apply a Schur complement decomposition.

Eliminating each W_k from (4.10) we get the following expression for $\Delta\Pi$,

$$\left[\delta_x \mathbb{I} - \sum_{k=1}^{N_S} A_k^T (W_k)^{-1} A_k \right] \Delta\Pi = r_\Pi - \sum_{k=1}^{N_S} A_k^T (W_k)^{-1} r_k \quad (4.11)$$

which requires forming the Schur complement, $B = \delta_x \mathbb{I} - \sum_{k=1}^{N_S} A_k^T (W_k)^{-1} A_k$, and solving this dense symmetric linear system for $\Delta\Pi$. Once a value for $\Delta\Pi$ is known, the remaining variables can be found by solving the following systems,

$$W_k \Delta v_k = r_k - A_k \Delta\Pi \quad (4.12)$$

for each $k = 1, \dots, N_S$. Note that in this strategy, the factorization of W_k and the solution of (4.12) can be performed independently in different processors. The Schur complement decomposition strategy applies specifically to the solution of the *augmented system* within the overall IPOPT algorithm and simply replaces the general default linear solver. The sequence of steps in the overall IPOPT algorithm are not altered, and as such, this specialized strategy *inherits all of the convergence properties* of the IPOPT algorithm for general nonlinear programs.

The decomposition strategy is straightforward to parallelize with excellent scaling properties. With $M = \dim(\Pi)$, the number of parameters, the number of linear solves of the W_k blocks required by the decomposition approach is $N \cdot M + 2N_S$. If the number of available processors in a distributed cluster is equal to N_S (one processor for each scenario), then the number of linear solves required by each processor is only $M + 2$, independent of the number of scenarios. This implies an approach that scales well with the number of scenarios. As we increase the number of scenarios under consideration, the cost of the linear solve remains fairly constant (with minimal communication overhead) as long as an additional processor is available for each new scenario. More importantly, the memory required on each processor is also nearly constant, allowing us to expand the number of scenarios and, using a

large distributed cluster, move beyond the memory limitation of a standard single processor machine. In Chapter 8 we demonstrate the computational efficiency of this approach for a large-scale estimation problem arising in a low-density polyethylene tubular reactor process.

4.3.2 Interpretation of Estimates

It is usually not enough to obtain an optimal value of the parameters. We must also investigate their reliability and precision. For instance, it is desired to answer questions such as "How much can we change the parameters and still match the data?", "What is the effect of data errors on the parameters?". This can be done by analyzing variations of the estimated parameters in the presence of variations of the measurement data. A parameter is usually said to be ill-conditioned if its estimated value is affected strongly by small variations on the data [7]. Here, it is said that the parameter estimates have a large variance which is reflected in the so-called *covariance matrix*.

Note that the optimal value of the parameters is an implicit function of the problem data (measurements). Following the notation used in the NLP sensitivity section 3.2.2 and the inference analysis of Bard [7], we can represent the KKT conditions of the standard least-squares parameter estimation problem (4.1) as,

$$\varphi(\Pi(\eta), \eta) = 0. \quad (4.13)$$

For simplicity in the presentation and without loss of generality, we have assumed that the parameters Π are the only degrees of freedom of the problem. In the spirit of the results of section (3.2.2), we apply the implicit function theorem around an optimal nominal solution $\Pi_*(\eta_0)$,

$$\bar{\mathbf{K}}_*(\eta_0) \frac{\partial \Pi_*}{\partial \eta} = - \frac{\partial \varphi(\Pi_*(\eta_0), \eta_0)}{\partial \eta} \quad (4.14)$$

so that,

$$\frac{\partial \Pi_*}{\partial \eta} = -\bar{\mathbf{K}}_*(\eta_0)^{-1} \left(\frac{\partial \varphi(\Pi_*(\eta_0), \eta_0)}{\partial \eta} \right). \quad (4.15)$$

If Π are the only degrees of freedom and no bounds are active at the solution, $\bar{\mathbf{K}}_*(\eta_0)$ represents the reduced Hessian matrix. The covariance matrix of the parameters is defined as,

$$\mathbf{V}_\Pi := E(\delta\Pi_*\delta\Pi_*^T). \quad (4.16)$$

where $E(\cdot)$ is the expectation operator and $\delta\Pi_*$ is the deviation of the *optimal* parameters for a given perturbation on the data from the current nominal solution. Computing the exact covariance matrix would require to *solve* a large number of parameter estimation problems for random variations of the data. To avoid this, we can also use the sensitivity matrix to compute *approximate* estimates of variations $\delta\Pi_*$ with respect to $\delta\eta$,

$$\delta\Pi_* \approx \frac{\partial\Pi_*}{\partial\eta}\delta\eta \quad (4.17)$$

and,

$$\mathbf{V}_\Pi \approx E\left(\frac{\partial\Pi_*}{\partial\eta}\delta\eta\delta\eta^T\frac{\partial\Pi_*}{\partial\eta}^T\right). \quad (4.18)$$

Since the sensitivity matrix is a constant evaluated at the current solution, it can be taken out of the expectation operator. This gives,

$$\mathbf{V}_\Pi \approx \frac{\partial\Pi_*}{\partial\eta}\mathbf{V}_\eta\frac{\partial\Pi_*}{\partial\eta}^T \quad (4.19)$$

where $\mathbf{V}_\eta = E(\delta\eta\delta\eta^T)$ is the covariance of the measurements (e.g. \mathbf{V}_y in (4.1)). If the variance of the measurements is *small and known* it is possible to prove that,

$$\mathbf{V}_\Pi \approx \bar{\mathbf{K}}_*(\eta_0)^{-1}. \quad (4.20)$$

In other words, the covariance of the parameters can be approximated by the inverse of the reduced Hessian matrix evaluated at the current solution. This result is of practical interest because we can extract this information directly from IPOPT. This analysis implies that, if the reduced Hessian has large eigenvalues, then the optimum is a sharp minimum as shown in Figure 4.1. Equivalently, the eigenvalues of the covariance matrix are small and the parameters are reliable. Furthermore, the inverse of the reduced Hessian becomes a better approximation of the true covariance matrix. As the optimum becomes more and

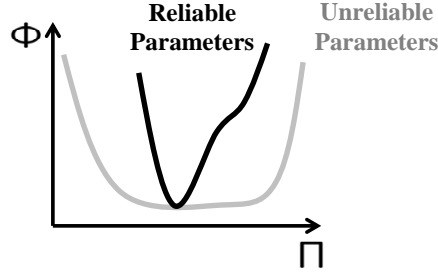


Figure 4.1: Schematic representation of the relationship between the reduced Hessian and the reliability of the estimated parameters.

more flat the reduced Hessian becomes nearly singular and the covariance matrix becomes infinite (poor reliability of the parameters). In addition, the inverse of the reduced Hessian becomes a poor approximation of the true parameter covariance matrix. This analysis also implies that, as we add more and more data sets into the estimation problem, then we can capture the variations of the measurements errors more accurately and thus decrease the covariance of the parameter estimates.

Note that when the solution of the estimation problem is a well-defined optimum it will satisfy the strong-second-order conditions (SSOC). SSOC in turn implies that the degrees of freedom of the NLP can be computed *uniquely*. As we discussed in Chapter 3, when the solution of the NLP satisfies SSOC, the KKT matrix will not need regularization. In the context of parameter estimation we can then guarantee that: *If no regularization of the KKT matrix is required at the solution, the estimated parameters have been uniquely determined (are observable)* [11, 125]. If the parameters cannot be uniquely determined, then the estimation problem is said to be *ill-posed*. From a practical point of view, this implies that the measurements are not informative enough.

Having the covariance matrix of the parameters, we can quantify their reliability through a confidence region analysis. Consider a quadratic approximation (Taylor series expansion)

of the objective function of (4.1) around the nominal solution,

$$\frac{1}{2}(\Pi - \Pi_*)^T \mathbf{V}_\Pi^{-1}(\Pi - \Pi_*) \approx \Phi(\Pi) - \Phi(\Pi_*) \leq \epsilon. \quad (4.21)$$

Contours of constant objective function values (i.e. constant ϵ) are ellipsoids defined by the parameters covariance. It can be shown that, if we have a large number data sets and if the measurement errors are independent and normally distributed with known covariance $\mathbf{V}_y = \text{diag}(\sigma^2)$, then ϵ follows a chi-square distribution $\chi^2(n_\Pi, \alpha)$ with n_Π degrees of freedom and confidence level α . This gives,

$$\frac{1}{2}(\Pi - \Pi_*)^T \mathbf{V}_\Pi^{-1}(\Pi - \Pi_*) \leq \sigma^2 \chi^2(n_\Pi, \alpha). \quad (4.22)$$

One can use this relationship to obtain *order-of-magnitude* estimates of the parameter confidence levels under different scenarios of practical interest. Note, however, that this inference analysis strategy only applies to standard least-squares formulations. For a description of more general inference techniques including EVM formulations please refer to [7, 2].

Illustrative Example

We consider a simple estimation problem of the form,

$$\min_{\Pi_1, \Pi_2, \Pi_3} \sum_{k=1}^{N_S} \frac{1}{2\sigma^2} (y_k - y_k^M)^2 \quad (4.23a)$$

$$\text{s.t. } y_k = \Pi_1 u_k + \Pi_2 u_k^2 + \Pi_3 u_k^3, \quad k = 1, \dots, N_S \quad (4.23b)$$

Using a set of fixed u_k , we use the polynomial model with the true parameter values $\Pi = [1 \ 2 \ 3]^T$ to generate the outputs y_k^M which are also corrupted by zero-mean Gaussian noise with variance $\sigma^2 = 0.001$. For a 95% confidence level and $n_\Pi = 3$ degrees of freedom we have $\sigma^2 \chi^2(n_\Pi, \alpha) = 0.001 \times 7.81$. We solve the estimation problem for different number of data sets using AMPL and IPOPT. We extract the inverse of the reduced Hessian (covariance matrix of parameters) using the previously proposed strategy.

Since the model is a third order polynomial, a problem with $N_S < 3$ is ill-posed and gives a singular reduced Hessian at the solution (regularization required by IPOPT). For

$N_S > 3$, we plot the ellipsoidal confidence region for the pair Π_2, Π_3 in Figure 4.2. The outer ellipsoid corresponds to $N_S = 3$ while the inner ellipsoid corresponds to $N_S = 10$. As expected, the confidence regions shrink as we add information into the problem. Note also that the confidence regions are not concentric due to the model nonlinearity.

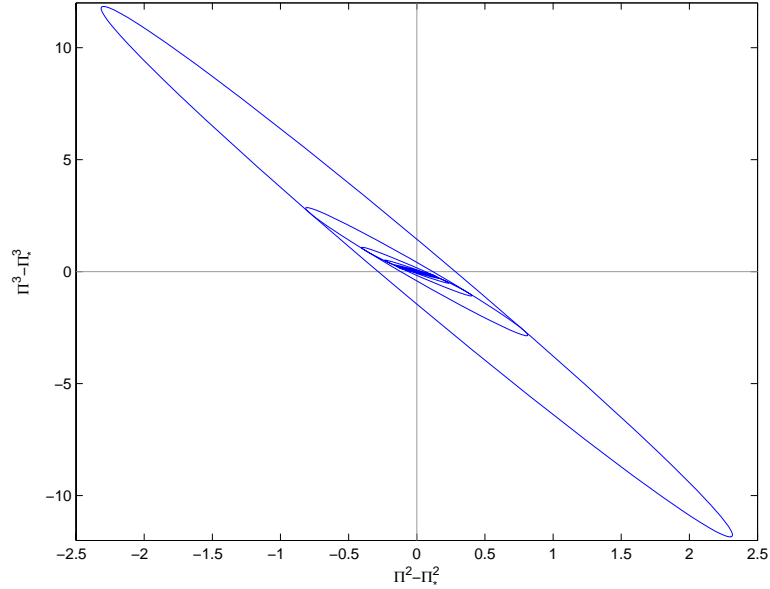


Figure 4.2: Ellipsoidal confidence regions for polynomial parameter estimation example.

4.4 Concluding Remarks

In this chapter, we have presented computational strategies for the solution of large-scale parameter estimation problems arising in off-line model development tasks. We discuss how to formulate multi-set parameter estimation problems as multi-scenario NLPs. We see that this can be used to exploit the inherent structure of the problem in parallel computer architectures to obtain fast solutions and to avoid computer memory bottlenecks.

We discuss how to verify for the uniqueness of the parameter estimates from the inertia

of the KKT matrix. Finally, we demonstrate that it is possible to use reduced Hessian information extracted from IPOPT to perform large-scale inference analysis in complex estimation problems.

In the next chapter, we move from off-line model development tasks to on-line optimization tasks arising in operations. In particular, we discuss computational strategies for Nonlinear Model predictive Control (NMPC).

Chapter 5

Nonlinear Model Predictive Control

In this chapter, we analyze computational issues associated with Nonlinear Model Predictive Control (NMPC) and propose strategies to overcome them. In particular, we derive an advanced-step strategy that computes fast on-line approximate solutions using NLP sensitivity concepts. We demonstrate that these approximations are accurate and do not deteriorate significantly the stability properties of the NMPC controller.

5.1 Introduction

NMPC is a model-based feedback control strategy based on the on-line solution of moving horizon optimal control problems (OCPs). This approach has the key advantage that it is a *general* multivariable control strategy that can handle constrained, nonlinear systems directly. NMPC has evolved over the past decade into an efficient method for the control of large industrial systems. In a recent IFAC workshop, several industrial applications were presented including contributions from ExxonMobil [10], BASF [91] and ABB [48] demonstrating the effectiveness of this method.

In addition to enabling advances in DAE-constrained optimization, the evolution of NMPC has been due to a much better understanding of its stability and robustness properties [86]. Moreover, with the ability to solve DAE-constrained optimization problems

on-line, the separation between model predictive control and real-time optimization will start to disappear (as in D-RTO). On the other hand, the solution of the NMPC problem is a time-critical step that puts demands on more powerful algorithms and their implementations. The need to consider larger applications leads to the challenging task of maintaining controller stability and performance in the face of *on-line computational delays*.

Performance deterioration of NMPC due to computational delays has been noted by Santos et al. in a laboratory reactor [108] as well as in numerous industrial studies. Deterioration of stability has been studied in [42, 28]. To address this issue, real-time NMPC strategies such as explicit schemes, neighboring extremals, Newton-type controllers and NLP sensitivity-based controllers represent some alternatives. Explicit schemes compute off-line control actions based on a full enumeration of possible states. This approach is most suitable for systems with a few states where the effect of combinatorics is rather small [12, 53]. For systems with large state spaces, on-line NMPC controllers represent a more efficient alternative. Among these, Newton-type controllers perform a single iteration (full Newton step) in the solution of the OCP at each time step. This requires the solution of a quadratic programming (QP) problem constructed around the solution of the QP at the previous time step [81, 96, 35, 32]. This allows a fast disturbance rejection mechanism that has shown good practical performance in some applications. In addition, it can be shown that if the series of QPs is initialized around a sufficiently good reference solution, then the QP series can converge to the solution of the moving horizon OCPs. This result follows from the local convergence properties of Newton's method and the parametric properties of the moving horizon OCPs. With this, nominal stability of Newton-type controllers can be guaranteed in the face of this approximation by making use of the inherent robustness properties of NMPC [36]. Newton-step strategies based on both single and multiple-shooting approaches have been previously proposed.

The local convergence properties of Newton-type strategies might deteriorate in the face of nonlinear effects and strong perturbations, thus requiring extra safeguards to promote

global convergence [81, 33]. A strategy able to ameliorate this drawback relies on the construction of approximate solutions around a continuously updated reference solution. Different variants of this are based on neighboring extremals [21, 98, 95] and NLP sensitivity concepts [26, 25, 64, 131]. The nominal stability results of Newton-type controllers can also be used for these controller variants. However, a more general and constructive analysis of their stability properties can become cumbersome under these arguments, thus leading to unnecessarily conservative assumptions.

Motivated by these observations, we propose an advanced-step NMPC (asNMPC) controller formulation [126]. The idea is to use the current control action to predict the future plant state in order to solve the future OCP in advance, while the current sampling period evolves. In the nominal case, the prediction matches the future plant state so that the current solution is already available, thus avoiding the computational delay. Similar ideas have been previously proposed in [42, 28]. In this case, the asNMPC controller inherits the same nominal stability properties of the ideal NMPC (iNMPC) controller. An issue associated with this strategy is treatment of disturbances and model mismatch. To account for these, the proposed controller exploits the parametric property of the OCP and approximates the true optimal solution using NLP sensitivity calculations. A direct consequence of this is that a rigorous bound on the loss of optimality can be established and related to the bounds of the uncertainty description. With this, it is possible to analyze the inherent robustness properties of the asNMPC controller in a straightforward manner using input-to-state stability concepts. We show that the resulting closed-loop system is input-to-state stable and contrast its stability bounds with those of the iNMPC controller. Here, we demonstrate that NLP sensitivity errors do not deteriorate significantly the stability properties of the asNMPC controller. While the proposed controller is applicable to any solution approach to DAE-constrained optimization problems, the advanced-step strategy is most suitable for the full discretization approach since this can provide fast background solutions. We illustrate the concepts using a classical nonlinear CSTR example and discuss computational

issues associated with larger scale systems.

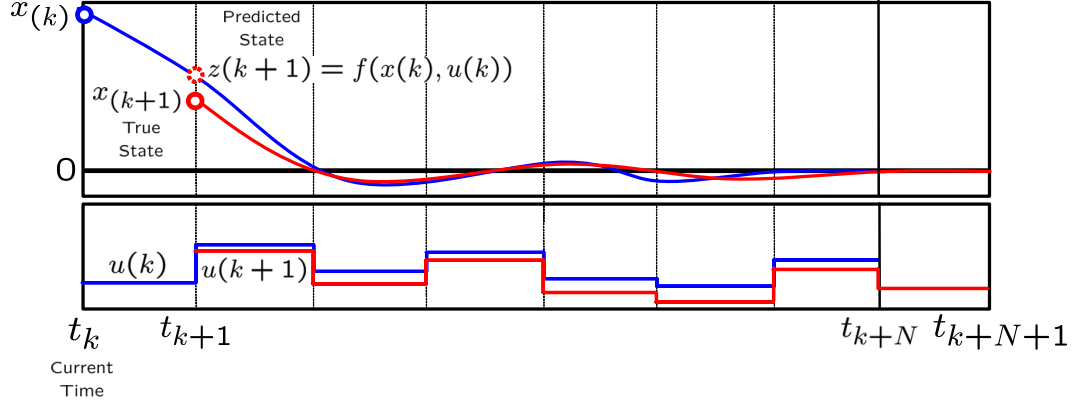


Figure 5.1: Typical NMPC scenario.

5.2 Standard NMPC Formulation

Consider the scenario sketched in Figure (5.1). We assume that the dynamics of an *uncertain* plant can be described by the following discrete-time dynamic model,

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), u(k), w(k)) \\ &= f(x(k), u(k)) + g(x(k), u(k), w(k)) \end{aligned} \quad (5.1)$$

where $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$ and $w(k) \in \mathbb{R}^{n_w}$ are the plant states, controls and disturbance signals, respectively, defined at time steps t_k with integers $k > 0$. The mapping $f : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}^{n_x}$ with $f(0,0) = 0$ represents the nominal model,

$$x(k+1) = f(x(k), u(k)) \quad (5.2)$$

while the term $g : \mathbb{R}^{n_x+n_u+n_w} \mapsto \mathbb{R}^{n_x}$ can be used to describe modeling errors, estimation errors and disturbances, among others. Having $x(k)$, the current plant state or an estimate, NMPC uses the nominal model,

$$z_{l+1} = f(z_l, v_l), \quad z_0 = x(k), \quad l = 0, \dots, N \quad (5.3)$$

to find a control sequence $\{v_0, v_1, \dots, v_N\}$ and associated state sequence $\{z_0, z_1, \dots, z_N\}$ that minimizes the cost function defined as,

$$J_N := F(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \quad (5.4)$$

over a future prediction horizon containing N time steps. Here, the computed controls $v_l \in \mathbb{R}^{n_u}$ and predicted states $z_l \in \mathbb{R}^{n_x}$ are enforced to satisfy the constraints $v_l \in \mathbb{U}$ and $z_l \in \mathbb{X}$ and the terminal constraints $z_N \in \mathbb{X}_f \subseteq \mathbb{X}, \forall l$. The cost function $J_N : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}$ comprises the stage costs $\psi : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}$ and a terminal penalty function $F : \mathbb{R}^{n_x} \mapsto \mathbb{R}$. This gives rise to a parametric NLP problem $\mathcal{P}_N(x(k))$ of the form,

$$\begin{aligned} \min_{v_l, z_l} \quad & J_N(\eta^{mpc}) := F(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \\ \text{s. t. } \quad & z_{l+1} = f(z_l, v_l), \quad z_0 = x(k) \quad l = 0, \dots, N-1 \\ & z_l \in \mathbb{X}, \quad z_N \in \mathbb{X}_f, \quad v_l \in \mathbb{U}. \end{aligned} \quad (5.5)$$

The solution of $\mathcal{P}_N(x(k))$, (z_l^*, v_l^*) , provides an *optimal cost value* $J_N(x(k))$ as well as the control $u(k) = v_0^*$ which is injected into the plant. In the nominal case, this drives the state of the plant towards $x(k+1) = z(k+1) = f(x(k), u(k))$ where $z(k+1) \in \mathbb{R}^{n_x}$ is the nominal model prediction. In the face of uncertainty, the plant evolves as in (5.1) and generates the mismatch $x(k+1) - z(k+1) = g(x(k), u(k), w(k))$. Once $x(k+1)$ is known, the prediction horizon is shifted forward by one sampling instant and problem $\mathcal{P}_N(x(k+1))$ is solved to find $u(k+1)$. This recursive strategy gives rise to the feedback law,

$$u(k) = h^{id}(x(k)) \quad (5.6)$$

with $h^{id} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_u}$. Here, we will assume that the control action resulting from this conventional NMPC formulation can be computed instantaneously and term this the ideal NMPC (iNMPC) controller.

Many different choices of the penalty function $F(\cdot)$ and of the terminal set \mathbb{X}_f have been proposed to guarantee stability of the nominal closed-loop system (5.6). Infinite horizon,

quasi-infinite horizon, zero-state terminal constraint and dual-mode control represent some alternatives [88]. In addition, since the recursive solution of the nominal problem $\mathcal{P}_N(x(k))$ provides a mechanism to react to disturbances in (5.1), the nominal feedback law (5.6) provides some inherent robustness. This holds true in many important cases, except in the presence of state constraints for \mathbb{X}, \mathbb{X}_f where a robust formulation of $\mathcal{P}_N(x(k))$ is required (e.g. minmax NMPC). Here, we focus on nominal NMPC schemes and study their inherent robustness properties through input-to-state stability concepts. For a comprehensive summary of robustness analysis and general robust design of discrete-time NMPC algorithms please refer to [86].

Remark: The discrete-time representation of the dynamic model (5.2) and of the NMPC problem (5.5) are only *conceptual*. In most practical applications the dynamic process model is solved as a general continuous-time DAE system. In particular, note that the structure of the NMPC problem resembles that of the multi-stage problem (2.3). As we have seen in Chapter 2, the multi-stage representation can be derived from the DAEs by applying a numerical quadrature scheme on the domain $[t_k, t_{k+1}]$ corresponding to two neighboring sampling times. Alternatively, we can integrate the dynamic model on this domain using a DAE solver.

5.2.1 Optimality Conditions

The moving horizon OCP $\mathcal{P}_N(\cdot)$ is parametric in the initial state. Accordingly, it is possible to define the data vector $\eta^{mpc} := x(k)$. The following results are based on the post-optimal analysis of solutions of parametric NMPC problems. Consequently, we will simplify the presentation by handling the inequality constraints on the domains $\mathbb{X}, \mathbb{X}_f, \mathbb{U}$ implicitly (e.g. by adding a logarithmic barrier function as in Chapter 3).

The Lagrange function associated with $\mathcal{P}_N(\eta^{mpc})$ is given by,

$$\begin{aligned} \mathcal{L} = & F(z_N(\eta^{mpc})) + \lambda_0^T(\eta^{mpc})(z_0(\eta^{mpc}) - \eta^{mpc}) \\ & + \sum_{l=0}^{N-1} [\psi(z_l(\eta^{mpc}), v_l(\eta^{mpc})) + \lambda_{l+1}^T(\eta^{mpc})(z_{l+1}(\eta^{mpc}) - f(z_l(\eta^{mpc}), v_l(\eta^{mpc})))] \end{aligned} \quad (5.7)$$

where $\lambda_l(\eta^{mpc}) \in \mathbb{R}^{n_x}$ are vectors of Lagrange multipliers. Note that all the primal and dual variables become implicit functions of η^{mpc} . For simplicity in the presentation, we suppress this argument from the notation. The solution of $\mathcal{P}_N(\eta^{mpc})$ needs to satisfy the first-order optimality or Karush-Kuhn-Tucker (KKT) conditions,

$$\left. \begin{aligned} \nabla_{\lambda_0} \mathcal{L} &= z_0 - \eta^{mpc} \\ \nabla_{z_l} \mathcal{L} &= \nabla_{z_l} \psi_l - A_l \lambda_{l+1} + \lambda_l = 0 \\ \nabla_{v_l} \mathcal{L} &= \nabla_{v_l} \psi_l - B_l \lambda_{l+1} = 0 \\ \nabla_{\lambda_{l+1}} \mathcal{L} &= z_{l+1} - f_l = 0 \\ \nabla_{z_N} \mathcal{L} &= \nabla_{z_N} F_N + \lambda_N = 0 \end{aligned} \right\} l = 0, \dots, N-1 \quad (5.8)$$

where $f_l := f(z_l, v_l)$, $\psi_l := \psi(z_l, v_l)$, $F_N := F(z_N)$, $A_l = \nabla_{z_l} f_l$ and $B_l = \nabla_{v_l} f_l$.

Using the NLP sensitivity results of Chapter 3, we want to study the effect of perturbations on η^{mpc} around an available nominal solution $s_*(\eta_0^{mpc}, N)$. In order to do this, we linearize the KKT conditions (5.8) around the nominal solution to give,

$$\left. \begin{aligned} \Delta z_0 &= (\eta^{mpc} - \eta_0^{mpc}) \\ Q_l \Delta z_l + W_l \Delta v_l - A_l^T \Delta \lambda_{l+1} + \Delta \lambda_l &= 0 \\ W_l^T \Delta z_l + R_l \Delta v_l - B_l^T \Delta \lambda_{l+1} &= 0 \\ \Delta z_{l+1} - A_l \Delta z_l - B_l \Delta v_l &= 0 \\ Q_N \Delta z_N + \Delta \lambda_N &= 0 \end{aligned} \right\} l = 0, \dots, N-1 \quad (5.9)$$

where $Q_N = \nabla_{z_N z_N} \mathcal{L} = \nabla_{z_N z_N} F$, $Q_l = \nabla_{z_l z_l} \mathcal{L}$, $W_l = \nabla_{z_l v_l} \mathcal{L}$ and $R_l = \nabla_{v_l v_l} \mathcal{L}$. In matrix form,

The computational complexity of this Riccati-like strategy scales as $O(N(n_z + n_u)^3)$ [103]. In other words, it is efficient for systems with few states and controls and long estimation horizons. Nevertheless, from this recursion it becomes clear that if we have an optimal control $v_0^*(\eta_0^{mpc})$ from $\mathcal{P}(\eta_0^{mpc})$, a fast approximate control action for a neighboring problem $\mathcal{P}(\eta^{mpc})$ can be obtained from the correction,

$$\tilde{v}_0(\eta^{mpc}) = v_0^*(\eta_0^{mpc}) + K_0 \cdot (\eta^{mpc} - \eta_0^{mpc}) \quad (5.13)$$

where K_0 is an analog of the Riccati gain matrix [21]. Note that the above correction step is equivalent to obtain $\tilde{v}_0(\eta^{mpc})$ from (3.36) or (3.33) with $\frac{\partial v_0^*}{\partial \eta^{mpc}} = K_0$.

5.2.2 Stability Properties

Ensuring stability of the closed-loop system (5.1)-(5.6) is a central problem in NMPC. Here, we establish well-known results on nominal and robust stability for the iNMPC controller. To start the discussion, we refer to [86, 68] to make use of the following definitions and assumptions:

Definition 6. A continuous function $\alpha(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is a \mathcal{K} function if $\alpha(0) = 0, \alpha(s) > 0, \forall s > 0$ and it is strictly increasing. A continuous function $\beta : \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{R}$ is a \mathcal{KL} function if $\beta(s, k)$ is a \mathcal{K} function in s for any $k > 0$ and for each $s > 0, \beta(s, \cdot)$ is decreasing and $\beta(s, k) \rightarrow 0$ as $k \rightarrow \infty$.

Definition 7. (Lyapunov function) A function $V(\cdot)$ is called an Lyapunov function for system (5.2) if there exist an invariant set \mathbb{X} , \mathcal{K} functions α_1, α_2 and α_3 such that, $\forall x \in \mathbb{X}$,

$$V(x) \geq \alpha_1(|x|) \quad (5.14a)$$

$$V(x) \leq \alpha_2(|x|) \quad (5.14b)$$

$$\Delta V(x) = V(f(x, h^{id}(x))) - V(x) \leq -\alpha_3(|x|) \quad (5.14c)$$

Assumption 1. (*Nominal Stability Assumptions of iNMPC*)

- The terminal penalty $F(\cdot)$, satisfies $F(z) > 0, \forall z \in \mathbb{X}_f \setminus \{0\}$,
- There exists a local control law $u = h_f(z)$ defined on \mathbb{X}_f , such that $f(z, h_f(z)) \in \mathbb{X}_f, \forall z \in \mathbb{X}_f$, and $F(f(z, h_f(z))) - F(z) \leq -\psi(z, h_f(z)), \forall z \in \mathbb{X}_f$.
- The optimal stage cost $\psi(x, u)$ satisfies $\alpha_p(|x|) \leq \psi(x, u) \leq \alpha_q(|x|)$ where $\alpha_p(\cdot)$ and $\alpha_q(\cdot)$ are \mathcal{K} functions.

Assumption 2. (*Computational Delay of iNMPC*) The control law $u = h^{id}(x)$ can be computed instantaneously.

Nominal stability of iNMPC can be paraphrased by the following theorem.

Theorem 5.2.1. (*Nominal Stability of iNMPC [88, 31]*) Consider the moving horizon problem $\mathcal{P}_N(x)$ defined in (5.5) and associated control law $u = h^{id}(x)$, that satisfies Assumptions 1 and 2. Then, $J_N(x)$ is a Lyapunov function and the closed-loop system is asymptotically stable.

Proof: We compare the costs of the neighboring problems $\mathcal{P}_N(x(k))$ and $\mathcal{P}_N(x(k+1))$. In the absence of uncertainty $x(k+1) = z(k+1) = f(x(k), h^{id}(x(k)))$ so that,

$$\begin{aligned}
 J_N(x(k+1)) - J_N(x(k)) &= J_N(z(k+1)) - J_N(x(k)) \\
 &\leq F(f(z_N^*, h_f(z_N^*))) - F(z_N^*) + \psi(z_N^*, h_f(z_N^*)) \\
 &\quad - \psi(x(k), u(k)) \\
 &\leq -\psi(x(k), u(k))
 \end{aligned}$$

The last two inequalities result from the fact that if a terminal controller satisfying Assumptions 1 exists, then the solution of $\mathcal{P}_N(x(k))$ provides a feasible solution to $\mathcal{P}_N(z(k+1))$ and the stage cost $\psi(x(k), u(k))$ is the only accumulation point. Since the stage cost is bounded by strictly positive functions, the cost function of the NMPC controller forms a

decaying sequence and becomes a Lyapunov function. By definition, the closed-loop system is asymptotically stable. \square

For the analysis of robust stability properties of the iNMPC controller, we apply definitions and properties of Input-to-State Stability (ISS) [86, 63].

Definition 8. (*Input-to-State Stability*) The system,

$$x(k+1) = \hat{f}(x(k), h^{id}(x(k)), w(k)), \quad k \geq 0, \quad x(0) = x_0 \quad (5.15)$$

is said to be ISS in \mathbb{X} if there exists a \mathcal{KL} function β , and a \mathcal{K} function γ such that for all $w \in \mathcal{W}$,

$$|x(k)| \leq \beta(|x_0|, k) + \gamma(|w|), \quad \forall k \geq 0, \quad \forall x_0 \in \mathbb{X} \quad (5.16)$$

Definition 9. (*ISS-Lyapunov function*) A function $V(\cdot)$ is called an ISS-Lyapunov function for system (5.15) if there exist a set \mathbb{X} , \mathcal{K} functions $\alpha_1, \alpha_2, \alpha_3$ and σ such that, $\forall x \in \mathbb{X}$ and $\forall w \in \mathcal{W}$,

$$V(x) \geq \alpha_1(|x|) \quad (5.17a)$$

$$V(x) \leq \alpha_2(|x|) \quad (5.17b)$$

$$\begin{aligned} \Delta V(x, w) &= V(\hat{f}(x, h^{id}(x), w)) - V(x) \\ &\leq -\alpha_3(|x|) + \sigma(|w|) \end{aligned} \quad (5.17c)$$

Lemma 5.2.2. [86, 63] Let \mathbb{X} be a robustly invariant set for system (5.15) that contains the origin and let $V(\cdot)$ be an ISS-Lyapunov function for this system, then the resulting system is ISS in \mathbb{X} .

To deal with robustness of the controller, we recognize that given $u(k)$ and the nominal model prediction $z(k+1) = f(x(k), u(k))$, there will exist a future mismatch $x(k+1) - z(k+1) = g(x(k), u(k), w(k))$ at the next time step, giving rise to two different problems $\mathcal{P}_N(z(k+1))$ and $\mathcal{P}_N(x(k+1))$, with optimal costs $J_N(z(k+1))$ and $J_N(x(k+1))$, respectively. To account for this, we define the mismatch term [109],

$$\epsilon(x(k+1)) := J_N(x(k+1)) - J_N(z(k+1)). \quad (5.18)$$

Assumption 3. Under Theorem 3.2.2, there exists a local positive Lipschitz constant L_J such that $\forall x \in \mathbb{X}$,

$$|\epsilon(x(k+1))| \leq L_J |g(x(k), u(k), w(k))|. \quad (5.19)$$

Assumption 4. (Robust Stability Assumptions) For $u = h^{id}(x)$,

- $g(x, u, w)$ can be described by \mathcal{K} functions so that: $|g(x, u, w)| \leq \alpha_g(|x|) + \sigma(|w|)$.
- there exists a \mathcal{K} function α_4 and all $w \in \mathcal{W}$ and there exists a constant $M > 0$ such that:

$$-\psi(x, u) + M(\alpha_g(|x|) + \sigma(|w|)) \leq -\alpha_4(|x|) + \sigma(|w|).$$

Robust stability of the iNMPC controller can be established from the following theorem.

Theorem 5.2.3. (Robust ISS Stability of iNMPC [86, 63]) Under Assumptions 1 and 4, with $M \geq L_J$, the cost function $J_N(x)$ obtained from the solution of $\mathcal{P}_N(x)$, is an ISS-Lyapunov function and the resulting closed-loop system is ISS stable.

Proof: We compare the costs of the neighboring problems $\mathcal{P}_N(x(k))$ and $\mathcal{P}_N(x(k+1))$ and introduce the effect of disturbances through $\epsilon(x(k+1))$,

$$\begin{aligned} J_N(x(k+1)) - J_N(x(k)) &= J_N(z(k+1)) - J_N(x(k)) + J_N(x(k+1)) - J_N(z(k+1)) \\ &\leq F(f(z_N^*, h_f(z_N^*))) - F(z_N^*) + \psi(z_N^*, h_f(z_N^*)) \\ &\quad - \psi(x(k), u(k)) + \epsilon(x(k+1)) \\ &\leq -\psi(x(k), u(k)) + \epsilon(x(k+1)) \end{aligned}$$

Again, the last two inequalities result from the fact that the solution of $\mathcal{P}_N(x(k))$ provides a feasible solution to $\mathcal{P}_N(z(k+1))$ and the stage cost $\psi(x(k), u(k))$ is the only accumulation point. Making use of Assumptions 3 and 4 to bound the mismatch term leads to,

$$J_N(x(k+1)) - J_N(x(k)) \leq -\alpha_4(|x(k)|) + \sigma(|w(k)|)$$

which fulfills (5.17c) with $M \geq L_J$. From Assumption 1 we also know that $J_N(x(k))$ satisfies (5.17a)-(5.17b). Therefore, the desired result follows from Lemma 5.2.2. In the nominal case, stability follows with $M = 0$. \square

Corollary 5.2.4. (*Asymptotic Robust Stability [86]*) *Assume that $\mathcal{W} = \{0\}$ and that Assumptions 1 -4 hold. Then the closed-loop system given by (5.1) and $u = h^{id}(x)$ is asymptotically stable.*

Remark: The above results assume Lipschitz continuity of the optimal cost function and of the control law. For the general nonlinear systems considered in this work, we guarantee Lipschitz continuity in a restricted neighborhood (possibly small) of an optimal solution satisfying the conditions of Theorem 3.2.2.

5.2.3 Computational Issues

It is clear that Assumption 2 is too restrictive since in practical applications $\mathcal{P}_N(x)$ may be computationally expensive to solve. This implies that the control action $u = h^{id}(x)$ cannot be injected into the plant right after x is obtained but only once the solution of $\mathcal{P}_N(x)$ has been obtained. As a consequence, the resulting delayed feedback action will be inconsistent with the current evolving state [108, 42, 28].

5.3 Advanced-Step Formulation

Consider that the state of the plant at t_k is $x(k)$ and that we know the control action $u(k)$. In the nominal case, the system evolves as in (5.2). As a consequence, starting at t_k we can predict the future state $z(k+1)$ and solve the predicted problem $\mathcal{P}_N(z(k+1))$ in advance. If this problem can be solved during the current sampling time, then $u(k+1) = h^{id}(x(k+1))$ will already be available at t_{k+1} . This simple strategy allows to remove the computational delay and preserves the iNMPC controller properties. In the presence of disturbances,

the plant will evolve with uncertain dynamics towards the true state $x(k+1) = z(k+1) + g(x(k), u(k), w(k))$. In this case, the iNMPC control action cannot be computed in advance. In order to account for this, we exploit the parametric property of the OCP problem to compute a fast NLP sensitivity approximate solution of $\mathcal{P}_N(x(k+1))$ around the available nominal solution of $\mathcal{P}_N(z(k+1))$ to obtain fast feedback. We call the resulting algorithm the advanced-step NMPC controller (asNMPC):

In background, between t_k and t_{k+1} :

- Having $x(k)$ and $u(k)$, predict the future state through forward simulation $z(k+1) = f(x(k), u(k))$. Set $\eta_0^{mpc} = z(k+1)$ and solve the *predicted* problem $\mathcal{P}(\eta_0^{mpc})$.
- At the solution $s^*(\eta_0^{mpc}, N)$, retain factors of $\mathbf{K}^*(\eta_0^{mpc}, N)$ or compute sensitivity matrix $\frac{\partial s^*}{\partial \eta^{mpc}}$ from (3.32).

On-line, at t_{k+1} :

- Obtain the true state $x(k+1)$ and set $\eta^{mpc} = x(k+1)$. Compute the fast approximate solution $\tilde{s}(\eta^{mpc}, N)$ from sensitivity (3.33) or as a perturbed Newton step (3.36), extract $u(k+1) = \tilde{v}_0(x(k+1))$ and return to background.

The above asNMPC algorithm yields the approximate control law,

$$u(k) = h^{as}(x(k)) \quad (5.20)$$

with the following property,

Theorem 5.3.1. (*Error Bound of asNMPC*) From Theorem 3.2.2 with $\eta_0^{mpc} = z(k+1)$ and $\eta^{mpc} = x(k+1) = z(k+1) + g(x(k), u(k), w(k))$, the approximation error between the asNMPC and iNMPC control laws satisfies $|h^{as}(x(k+1)) - h^{id}(x(k+1))| \leq L_h^{as} |g(x(k), u(k), w(k))|^2$ with a local positive Lipschitz constant L_h^{as} .

Proof: The asNMPC control action $\tilde{v}_0(x(k+1))$ is extracted from the approximate solution $\tilde{s}(x(k+1))$ obtained from the perturbation $\eta^{mpc} - \eta_0^{mpc} = g(x(k), u(k), w(k))$. From (5.13) we have,

$$\begin{aligned} u(k+1) &= \tilde{v}_0(x(k+1)) \\ &= v_0^*(z(k+1)) + K_0 \cdot (x(k+1) - z(k+1)) \\ &= v_0^*(z(k+1)) + K_0 \cdot g(x(k), u(k), w(k)). \end{aligned} \quad (5.21)$$

The error bound follows from (3.34) and the equivalence between (5.13), (3.33) and (3.36) to give,

$$|u(k+1) - v_0^*(x(k+1))| \leq L_h^{as} |g(x(k), u(k), w(k))|^2. \quad \square$$

For later reference, we note that solving the forward problem $\mathcal{P}_N(z(k+1)) = \mathcal{P}_N(f(x(k), h^{as}(x(k))))$ is equivalent to solve the following *extended* problem $\mathcal{P}_{N+1}(x(k), h^{as}(x(k)))$,

$$\begin{aligned} \min_{v_l, z_l} \quad J_{N+1} &:= F(z_N) + \psi(x(k), h^{as}(x(k))) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \\ \text{s. t. } z_{l+1} &= f(z_l, v_l), \quad l = 0, \dots, N-1 \\ z_0 &= f(x(k), h^{as}(x(k))) \\ z_l &\in \mathbb{X}, \quad z_N \in \mathbb{X}_f, \quad v_l \in \mathbb{U}. \end{aligned} \quad (5.22)$$

with fixed $h^{as}(x(k))$ computed from (5.21). In the following section, we will see that the cost function $J_{N+1}(x, h^{as}(x))$ associated with this problem can be used as a candidate Lyapunov function to derive sufficient stability conditions for the asNMPC control law.

5.3.1 Stability Properties

To analyze the stability properties of the proposed controller we make use of the assumptions and definitions of the standard NMPC formulation with a slight modification,

Assumption 5. (*Computational Delay of asNMPC*) *The background calculations associated with the solution of the forward problem $\mathcal{P}_N(f(x, h^{as}(x)))$ can be obtained in one*

sampling time. Moreover, the sensitivity update can be obtained in a negligible amount of time.

In the nominal case, the asNMPC and iNMPC controllers produce identical control actions. This follows from Theorem 5.3.1 with $g(x, u, w) = 0$. Under Assumption 5, Theorem 5.2.1 applies directly. For the analysis of the robustness properties of the asNMPC controller it is necessary to account for the effect of NLP sensitivity errors.

As shown in Figure 5.2, we recognize that the forward simulation $z(k+1) = f(x(k), u(k))$ will predict the future state at t_{k+1} . In the nominal case, this would give rise to the control action $h^{id}(z(k+1)) = h^{as}(z(k+1))$ that would be used to start the extended problem $\mathcal{P}_{N+1}(z(k+1), h^{id}(z(k+1)))$ with cost $J^{id}(z(k+1)) := J_{N+1}(z(k+1), h^{id}(z(k+1)))$.

In the robust case, the plant will evolve with uncertain dynamics generating $x(k+1)$. For the ideal NMPC controller this would give rise to the optimal control action $h^{id}(x(k+1))$ that would be used to solve $\mathcal{P}_{N+1}(x(k+1), h^{id}(x(k+1)))$ at the next time step with cost $J^{id}(x(k+1)) := J_{N+1}(x(k+1), h^{id}(x(k+1)))$. In reality, we compute the approximate control $h^{as}(x(k+1))$ from (5.21) giving rise to problem $\mathcal{P}_{N+1}(x(k+1), h^{as}(x(k+1)))$ with cost $J^{as}(x(k+1)) := J_{N+1}(x(k+1), h^{as}(x(k+1)))$. This is a suboptimal cost that needs to be compared against the optimal cost $J^{id}(x(k+1))$. To account for this, we define the following mismatch terms,

$$\epsilon_s(x(k+1)) := J^{id}(x(k+1)) - J^{id}(z(k+1)) \quad (5.23a)$$

$$\epsilon_{as}(x(k+1)) := J^{as}(x(k+1)) - J^{id}(x(k+1)) \quad (5.23b)$$

where the first term accounts for the model mismatch as in (5.18) while the second term accounts for approximation errors introduced by NLP sensitivity.

Assumption 6. *Under Theorems 3.2.2 and 5.3.1 there exist positive Lipschitz constants*

L_J, L_h and L_h^{as} such that $\forall x \in \mathbb{X}$,

$$\begin{aligned} |\epsilon_s(x(k+1))| &\leq L_J(|x(k+1) - z(k+1)| + |h^{id}(x(k+1)) - h^{id}(z(k+1))|) \\ &\leq L_J(1 + L_h)|g(x(k), u(k), w(k))| \end{aligned} \quad (5.24a)$$

$$\begin{aligned} |\epsilon_{as}(x(k+1))| &\leq L_J(|x(k+1) - x(k+1)| + |h^{as}(x(k+1)) - h^{id}(x(k+1))|) \\ &\leq L_J L_h^{as} |g(x(k), u(k), w(k))|^2. \end{aligned} \quad (5.24b)$$

By comparing the successive costs $J^{as}(x(k))$ and $J^{as}(x(k+1))$, we can arrive at a similar ISS property as in Theorem 5.2.3.

Theorem 5.3.2 (Robust Stability of asNMPC). *Under Assumptions 1, 4 and 5 with $M \geq L_J(1 + L_h + L_h^{as}|g(x, u, w)|) > 0$, the cost function $J^{as}(x)$ obtained from the solution of the extended problem $\mathcal{P}_{N+1}(x, u)$ with $u = h^{as}(x)$ is an ISS-Lyapunov function and the resulting closed-loop system is ISS stable.*

Proof : We compare the costs $J^{as}(x(k))$, $J^{as}(x(k+1))$ and use the mismatch terms in (5.24a)-(5.24b) to obtain,

$$\begin{aligned} J^{as}(x(k+1)) - J^{as}(x(k)) &= J^{id}(z(k+1)) - J^{as}(x(k)) \\ &\quad + J^{id}(x(k+1)) - J^{id}(z(k+1)) \\ &\quad + J^{as}(x(k+1)) - J^{id}(x(k+1)) \\ &\leq -\psi(x(k), h^{as}(x(k))) + \epsilon_s(x(k+1)) + \epsilon_{as}(x(k+1)) \end{aligned} \quad (5.25)$$

the last inequality results from noting that the solution of $\mathcal{P}_{N+1}(x(k), h^{id}(x(k)))$ provides a feasible solution to $\mathcal{P}_{N+1}(z(k+1), h^{id}(z(k+1)))$. Applying the bounds (5.24a)-(5.24b), the result follows with $M \geq L_J(1 + L_h + L_h^{as}|g(x(k), u(k), w(k))|) > 0$. \square

Note that if NLP sensitivity errors vanish (e.g. QP arising in linear MPC) then $\epsilon_{as}(x) = 0$. Accordingly, $M \geq L_J(1 + L_h)$ is sufficient and we recover *similar* (i.e. one step forward) robust stability properties of iNMPC as in Theorem 5.2.3. In the nominal case, $x(k+1) = z(k+1)$ and $M = 0$ is sufficient for nominal stability.

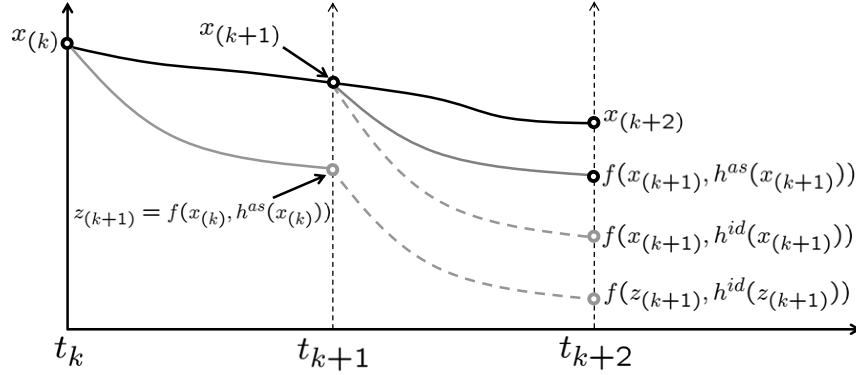


Figure 5.2: Schematic representation of Advanced-Step NMPC controller behavior.

5.3.2 Computational Issues

The proposed asNMPC algorithm is expected to reduce the on-line computational cost by two or three orders of magnitude. This results from the difference between the computational complexity of a single backsolve against that of the formation and factorization of the KKT matrix [131]. However, note that Assumption 5 requires that the background solution can be obtained in one sampling time. While a full-discretization strategy coupled to a full-space interior-point solver allows to obtain relatively fast background solutions, it is also necessary to derive an appropriate warm-start strategy to reduce the number of iterations. For NMPC problems, a well-known strategy consists on shifting the optimal values of the primal and dual variables of the current problem one step forward in time [103, 34]. For instance, if the solution of the current problem $\mathcal{P}_N(x(k))$ is given by

$$s_*(x(k)) = \{z_0, \dots, z_{N-1}, z_N, v_0, \dots, v_{N-2}, v_{N-1}, \lambda_0, \dots, \lambda_{N-1}, \lambda_N\}. \quad (5.26)$$

With this, we can generate the warm-start point $s_o(x(k))$ for problem $\mathcal{P}_N(x(k+1))$ as,

$$s_o(x(k+1)) = \{z_1, \dots, z_N, z_N, v_1, \dots, v_{N-1}, v_{N-1}, \lambda_1, \dots, \lambda_N, \lambda_N\} \quad (5.27)$$

Note that when the model is perfect then $z_1 = z(k+1) = x(k+1)$. In this case, as $N \rightarrow \infty$ then $s_o(x(k+1)) \rightarrow s_*(x(k+1))$. In other words, the warm-start point becomes

a better and better approximation of the solution. This result follows directly from the principle of optimality [34, 36]. On the other hand, as the model prediction deviates more and more from the actual plant state then the quality of the warm-start point deteriorates. In particular, if the perturbations are *sufficiently strong*, then the warm-start point will eventually be taken out of the region of attraction of Newton’s method where fast local convergence is expected [33, 93]. In this case, the NLP solver will require of appropriate globalization strategies (e.g. line-search, trust-region) to converge to the optimal solution. Note that the number of iterations required to solve the NMPC problem is in many cases problem dependent since it depends on the inherent nonlinearity of the model. However, it is important to recognize that there exists a complex interplay between the sampling time length, the time required per iteration and the predictive capabilities of the model. If this fact is recognized and properly exploited during early design stages of the controller, this can also be helpful in reducing the solution times.

In the context of the asNMPC controller there exists a direct connection between the quality of the NLP sensitivity approximations and the number of iterations required to solve the background problems. Based on the previous discussion we can see that, for *sufficiently* small perturbations, it is possible to use the NLP sensitivity approximations to generate the shifted warm-start points for the background problems.

Finally, note that the previous stability analysis assumes that the objective function of the NMPC problem is a convex least-squares function. Nevertheless, the *computational* advantages of the asNMPC controller are general.

5.4 Illustrative Example

We consider a simulated NMPC scenario on the CSTR described in Appendix C. The differential equations of the model are converted to the discrete-time form (5.2) through an orthogonal collocation of finite elements discretization scheme. Here, each finite element

represents a sampling time. To simulate the plant evolution, we introduce off-set free plant-model mismatch by perturbing the nominal value of the reactor residence time θ from its nominal θ_{nom} . In addition, we introduce Gaussian noise with σ standard deviation to the predicted states at each time step to simulate the presence of measurement or estimation errors coming from the plant. The noise is measured as a percentage on the nominal state values. The OCP is formulated using a quadratic function $\psi(z, v) = \hat{z}^T Q \hat{z} + R \hat{v}^2$ with $Q = \text{diag}\{1 \times 10^6, 2 \times 10^3\}$, $R = 1 \times 10^{-3}$, terminal weight $F(z) = \hat{z}^T Q \hat{z}$ and $\hat{z} = z - z_{ss}$, $\hat{v} = v - v_{ss}$ where subscript ss denotes steady-state value.

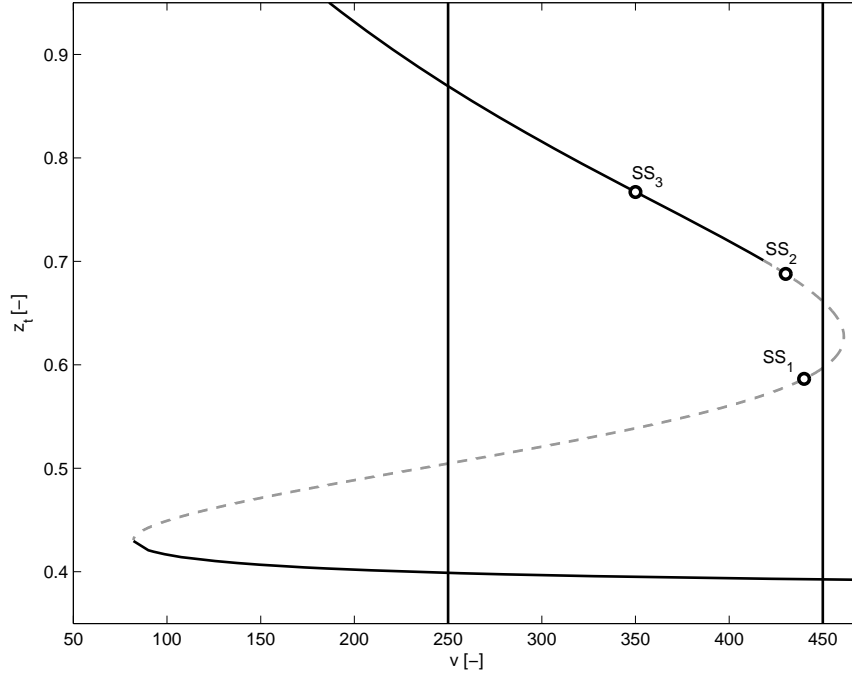


Figure 5.3: Steady-state map between temperature z^t and cooling water flow rate v . Solid vertical lines represent input constraints.

We contrast the performance of the iNMPC and asNMPC controllers under different robust scenarios. Here, we choose $N = 10$ time steps along with a zero terminal constraint $z_N = z_{ss}$. The resulting NLP problems contain $18 \times N$ variables and $6 \times N$ constraints.

The resulting NLPs can be solved in a negligible amount of time with IPOPT. In Chapter 8 we will see that the performance characteristics apply to much larger-scale systems as well. The controllers first perform the transition between two open-loop unstable steady states (SS_1 and SS_2) followed by a subsequent transition to a stable steady state, SS_3 . The location of the three steady-states is illustrated in the $v - z^t$ bifurcation diagram in Figure 5.3. The control is required to satisfy $250 \leq v \leq 450$ where the upper bound is set close to the corresponding steady-state values at SS_1 and SS_2 . We have seen that this tends to amplify approximation errors and thus illustrate the advantages and limitations of the proposed controller.

Scenario 1: In Figure 5.4 we illustrate the effect of increasing model mismatch due to perturbations on the reactor residence time. From the top graph it is clear that for a perturbation of ($\theta = 0.75\theta_{nom}$) the performance of both iNMPC and asNMPC is nearly identical. Both controllers are able to handle relatively large perturbations. However, as the mismatch is increased ($\theta = 0.5\theta_{nom}$) the performance of asNMPC tends to drift away and the closed-loop system destabilizes due to the presence of approximation errors. This is particularly evident in the second transition. Interestingly, for a slightly larger mismatch ($\theta = 0.45\theta_{nom}$) the iNMPC controller is not able to reject the perturbation in the second transition either, and the close-loop becomes unstable. In other words, both controllers are able to tolerate similar levels of mismatch, suggesting that the effect of approximation errors in the asNMPC controller is not very strong.

Scenario 2: Similar behavior can be seen when the controllers are subjected to simultaneous noise and model mismatch ($\theta = 0.75\theta_{nom}$) as illustrated in Figure 5.5. Again, for small levels of noise ($\sigma = 2.5\%$), the performance of the two controllers is almost identical. The asNMPC controller is able to tolerate large levels of noise (up to $\sigma = 7.5\%$) but its performance deteriorates due to approximation errors, specially in the transition from SS_1 to SS_2 .

Scenario 3: To illustrate the role of approximation errors on the stability of asNMPC,

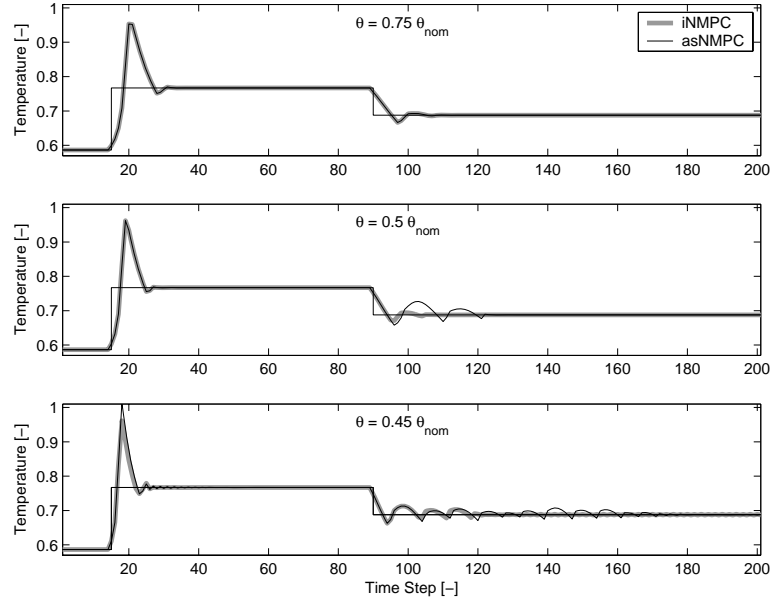


Figure 5.4: Scenario 1: Effect of plant-model mismatch on the performance of the controllers.

we perform a more detailed analysis on the second transition for scenario $\theta = 0.45\theta_{nom}$ from Figure 5.4. The results are illustrated in Figure 5.6. In the top graph, we present the profiles of the predicted $z(k)$ and the actual $x(k)$ temperatures. As can be seen, the perturbation in θ creates large deviations between both states. The mismatch is expected to generate a difference between the asNMPC control action $h^{as}(x)$ from that of the iNMPC $h^{id}(x)$ which is illustrated in the second graph. Interestingly, note that despite the relatively large mismatch, the asNMPC and iNMPC control actions are identical before the system destabilizes at time step 120. This would suggest that the system does not destabilize in the first place due to approximation errors. To validate this, we present profiles in the third graph of the left-hand side (*LHS*) and right-hand side (*RHS*) of (5.25), the sufficient stability condition from Theorem 5.3.2. Stability implies that $LHS \leq RHS$. As can be seen, this condition is fulfilled up to time step 95. However, note that even though the two control actions are identical at this point, there is a cross-over $LHS \geq$

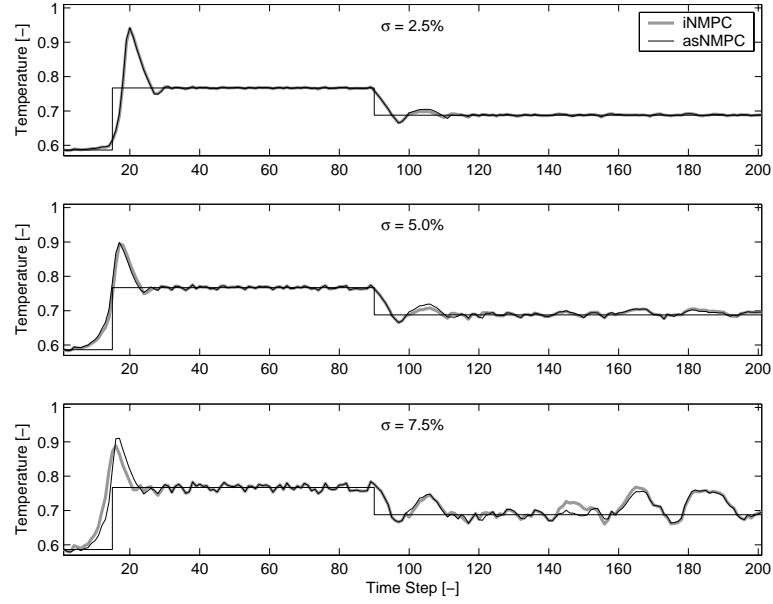


Figure 5.5: Scenario 2: Effect of noise on the performance of the controllers. Perturbation on residence time at 25% below nominal value.

RHS and the system destabilizes. To explain this, we present profiles for the mismatch terms in the bottom graph. As can be seen, the magnitude of the mismatch introduced by approximation errors $\epsilon_{as}(x)$ tends to be smaller compared to that introduced by the perturbations $\epsilon_s(x)$. However, at time step 95 the approximation errors become relevant and, *even though the injected control actions are identical*, the combined mismatch terms promote a cross-over in the stability condition (5.25) of Theorem 5.3.2. As predicted by (5.25), this destabilizes the system. It has been observed that, once the system becomes unstable, the oscillations become aggressive and the perturbations induce changes in the active-set for the perturbed problems (e.g. control profiles at time steps 120 and 140). These changes cannot be predicted by the NLP sensitivity calculation and require the solution of a quadratic programming problem (see Chapter 3). While this might improve the quality of the approximations, understanding the implications of active-set changes in the stability of the asNMPc controller requires a deeper analysis of the Lipschitz continuity assumptions

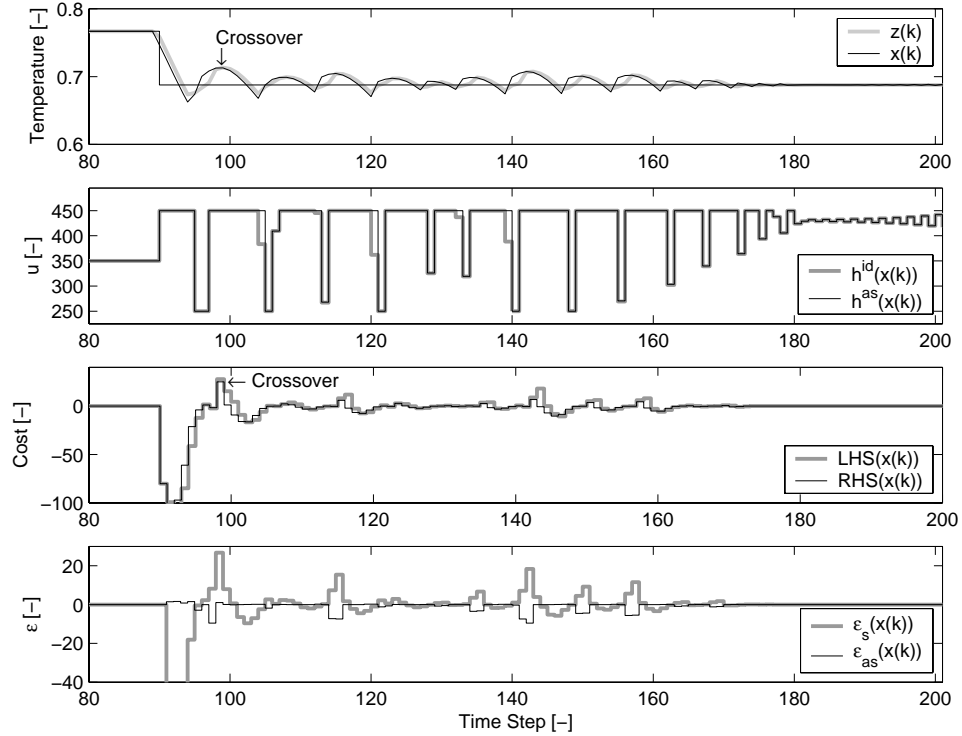


Figure 5.6: Scenario 3: Analysis of the effect of mismatch terms on the stability of the advanced step controller.

made in this work.

5.5 Concluding Remarks

In this chapter, we derive the advanced-step NMPC (asNMPC) controller and analyze its optimality and stability properties. The controller avoids feedback delays associated with the on-line solution of large-scale OCPs. Here, the moving horizon OCP is formulated with an advanced step control action and it is solved, in background, between sampling times. The results demonstrate that the asNMPC controller has identical nominal stability properties to the ideal NMPC controller without computational delay. In the presence of

disturbances, the controller exploits the parametric properties of the OCP through NLP sensitivity concepts to provide a fast on-line correction of the nominal solution. With this, a rigorous bound on the loss of optimality can be established and related to the bounds of the uncertainty description. This allows to characterize the robustness properties of the controller through input-to-state stability concepts. In addition, we establish connections of the asNMPC controller with classical Riccati-line regulators. In the following chapter, we derive advanced-step strategies for moving horizon state estimation.

Chapter 6

Moving Horizon Estimation

In this chapter, we discuss computational bottlenecks associated with on-line moving horizon estimation (MHE) problems. MHE is a powerful optimization-based strategy for state estimation but, as with NMPC, the on-line solution time becomes an issue in large-scale applications. We propose an advanced-step strategy to overcome these limitations. In addition, we present a detailed analysis of the KKT conditions of MHE problems to derive strategies to extract covariance information from general NLP solvers. Finally, we establish connections between the proposed advanced-step MHE estimator and traditional Kalman filters.

6.1 Introduction

The main idea behind state estimation is to use limited input and output measurement information to infer the state of the process and to obtain estimates of time-varying uncertain disturbances. This state information is required for the NMPC controller to compute the control actions and can also be used to obtain a deeper understanding of phenomena taking place in the process. In a general state estimation strategy, one starts with an *a priori* guess of the current state and associated statistics (e.g. covariance matrix) and use the model to predict the future state. At the next time step, measurement information is used to correct this prediction. We use the corrected state and statistics of the prior information at the next time step.

There exist different approaches to state estimation. Recursive approaches such as the

Kalman filter and its variations such as the Extended (EKF) and Unscented Kalman filter (UKF) are the most popular approaches in industry and academia [21, 104]. This has been mainly due to their practical simplicity and their solid foundations of probability theory. Many other strategies such as particle filters, Luenberger and recursive least-squares are also widely used. All these strategies present the common advantage that, if inequality constraints (bounds) are ignored, the estimation problem can be decomposed in a recursive manner (process one measurement at each time step). With this, they can avoid the solution of on-line optimization problems.

The difficulty to handle bound constraints is, on the other hand, an important limitation of recursive strategies since they can converge to regions of high probability but that are physically meaningless (e.g. negative concentrations) [57]. Optimization-based estimation strategies represent an important alternative to overcome these limitations [107, 116, 102, 104, 3]. Many of these strategies have evolved into what is known today as moving horizon estimation (MHE). In MHE, the estimator problem is formulated as an optimization problem incorporating batches of past measurement data. The ability to cast the estimator as a general optimization problem presents many advantages including the direct handling of bound constraints and the ability to handle sophisticated dynamic models in a computationally efficient manner. Different strategies have also been proposed to summarize past information in order to update the *a priori* state estimate and its statistics [102].

As in NMPC, a crucial limitation associated with MHE is that it requires on-line solutions of computationally intensive optimization problems. In particular, the solution time of the MHE problem affects the stability and performance of the NMPC as MHE provides the state estimates required by the controller. As with NMPC, Newton-step strategies for MHE have been previously proposed in [114, 43] in order to avoid these issues. In this section, we derive an advanced-step MHE (asMHE) strategy able to some of the limitations associated of Newton-based strategies [133]. In the spirit of the advanced-step NMPC controller, the main idea is to use the dynamic model to predict the future measurements. We use this

information to solve a predicted problem in between sampling times and correct this nominal solution on-line using an NLP sensitivity approximation. In addition, we derive strategies to extract covariance information from general NLP solvers and we establish connections between the proposed estimator and Kalman filters.

6.2 Standard MHE Formulation

Consider the scenario in which a given system is located at sampling time t_k and a past measurement sequence $\{y(k-N), \dots, y(k-1), y(k), u(k-N), \dots, u(k-1)\}$ is available. Here, $y(k)$ are the output measurements and $u(k)$ are the input measurements. We assume that the MHE estimator uses the *perfect* nonlinear model,

$$\begin{aligned} z_{l+1|k} &= f(z_{l|k}, w_{l|k}) = \hat{f}(z_{l|k}, u(k-N+l), w_{l|k}), \quad l = 0, \dots, N-1 \\ y(k-N+l) &= \chi(z_{l|k}) + v_{l|k}, \quad l = 0, \dots, N \end{aligned}$$

where $\chi(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is a nonlinear mapping between the states and outputs. The estimator uses the model to find the disturbance sequence $\{w_{0|k}, \dots, w_{N-1|k}\}$, the noise sequence $\{v_{0|k}, \dots, v_{N|k}\}$, and the initial state $z_{0|k}$ that minimizes the cost function,

$$\phi(\eta^{mhe}(k)) = \Gamma(z_{0|k}) + L_N(z_{N|k}) + \sum_{l=0}^{N-1} L_l(z_{l|k}, w_{l|k}). \quad (6.1)$$

with,

$$\begin{aligned} \Gamma(z_{0|k}) &= (z_{0|k} - \bar{z}_0(k))^T \bar{\Pi}_0^{-1}(k) (z_{0|k} - \bar{z}_0(k)) \\ L_l(z_{l|k}, w_{l|k}) &= (y(k-N+l) - \chi(z_{l|k}))^T R_l^{-1} (y(k-N+l) - \chi(z_{l|k})) + w_{l|k}^T Q_l^{-1} w_{l|k} \\ L_N(z_{N|k}) &= (y(k) - \chi(z_{N|k}))^T R_N^{-1} (y(k) - \chi(z_{N|k})) \end{aligned} \quad (6.2)$$

over a past horizon containing N time steps. Here, the computed disturbances $w_{l|k} \in \mathbb{R}^{n_w}$ and states $z_{l|k} \in \mathbb{R}^{n_z}$ are enforced to satisfy the constraints $w_{l|k} \in \mathbb{W}$ $z_{l|k} \in \mathbb{Z}$, respectively, $\forall l, k$.

The cost function $\phi : \mathfrak{R}^{Nnz+(N-1)nw} \rightarrow \mathfrak{R}$ contains a set of stage costs $L_l : \mathfrak{R}^{nz+nu} \rightarrow \mathfrak{R}$ and an initial penalty term $\Gamma : \mathfrak{R}^{nz} \rightarrow \mathfrak{R}$ summarizing prior information before t_{k-N} . Here, $\bar{z}_0(k)$ is the *a priori* state estimate with associated covariance $\bar{\Pi}_0(k)$. The matrices $\bar{\Pi}_0(k)$, Q_l , and R_l are assumed to be symmetric positive definite. In addition, we define the problem data,

$$\eta^{mhe}(k) = (\bar{z}_0(k), \bar{\Pi}_0^{-1}(k), y(k-N), \dots, y(k-1), y(k)) \quad (6.3)$$

that fully defines the current MHE problem. In compact form, this leads to an NLP of the form,

$$\begin{aligned} \mathcal{M}_N(\eta^{mhe}(k)) \quad \min_{z_{0|k}, w_{l|k}} \quad & \phi(\eta^{mhe}(k)) = \Gamma(z_{0|k}) + L_N(z_{N|k}) + \sum_{l=0}^{N-1} L_l(z_{l|k}, w_{l|k}) \\ \text{s.t.} \quad & z_{l+1|k} = f(z_{l|k}, w_{l|k}), \quad l = 0, \dots, N-1 \\ & z_{l|k} \in \mathbb{Z}, w_{l|k} \in \mathbb{W}. \end{aligned} \quad (6.4)$$

From the solution of this problem we obtain the optimal disturbance sequence $\{w_{0|k}^* \dots w_{N-1|k}^*\}$ and state sequence $\{z_{0|k}^* \dots z_{N|k}^*\}$ from which we extract the current estimate $\hat{x}^{id}(k) = z_{N|k}^*$ of the *true* plant state $x(k)$ with associated estimation error $e(k) := \hat{x}^{id}(k) - x(k)$.

At the next sampling time t_{k+1} , we get the new measurement $y(k+1)$ generated by the control $u(k)$ and define the new problem data $\eta^{mhe}(k+1)$. The measurements are shifted forward by one sampling time in order to drop the oldest measurement and include the current one. In addition, the *a priori* state estimate is updated by defining $\bar{z}_0(k+1) = z_{1|k}^*$. Finally, the associated covariance matrix is updated. As a standard practice, an EKF update is used [102],

$$\bar{\Pi}_0(k+1) = G_0 Q_0 G_0^T + A_0 \bar{\Pi}_0(k) A_0^T - A_0 \bar{\Pi}_0(k) C_0^T (R_0 + C_0 \bar{\Pi}_0(k) C_0^T)^{-1} C_0 \bar{\Pi}_0(k) A_0^T \quad (6.5)$$

where $A_0 = \nabla_z f(z_{0|k}^*, w_{0|k}^*)$, $G_0 = \nabla_w f(z_{0|k}^*, w_{0|k}^*)$, and $C_0 = \nabla_z \chi(z_{0|k}^*)$. It is important to note that $\bar{\Pi}_0(k+1)$ is only a linear *approximate* representation of the true covariance matrix. In addition, under certain conditions such as strong observability, the *a priori* covariance matrix does not need to be updated in order to achieve convergence of the estimator [3].

Having the updated problem data $\eta^{mhe}(k+1)$, a new MHE problem $\mathcal{M}_N(\eta^{mhe}(k+1))$ is solved to estimate the new state of the plant $x(k+1)$. In the following, if we assume that the MHE problems can be solved *instantaneously* at each sampling time, we refer to the resulting MHE algorithm as *ideal* MHE.

Remark: Note that the conceptual MHE formulation (7.3) implies that the input measurements $u(k)$ are fixed. Nevertheless, input errors can be modeled as disturbances $w(k)$. In addition, model parameters can be represented in this formulation by defining them as artificial states.

6.2.1 Optimality Conditions

The MHE problem $\mathcal{M}_N(\cdot)$ is parametric in the data η^{mhe} . We simplify the notation by dropping index k and adopt $z_{l|k} = z_l$, $w_{l|k} = w_l$, etc. The following results are based on the post-optimal analysis of solutions of parametric MHE problems. Consequently, we will simplify the presentation by handling the inequality constraints on the domains \mathbb{Z} and \mathbb{W} implicitly (e.g. add a logarithmic barrier function as in Chapter 3).

The Lagrange function associated with problem $\mathcal{M}(\eta^{mhe})$ is given by,

$$\begin{aligned} \mathcal{L} = & \Gamma(z_0(\eta^{mhe})) + L_N(z_N(\eta^{mhe})) + \sum_{l=0}^{N-1} L_l(z_l(\eta^{mhe}), w_l(\eta^{mhe})) \\ & + \sum_{l=0}^{N-1} \lambda_{l+1}^T(\eta^{mhe})(z_{l+1}(\eta^{mhe}) - f(z_l(\eta^{mhe}), w_l(\eta^{mhe}))) \end{aligned} \quad (6.6)$$

where $\lambda(\eta^{mhe}) \in \mathbb{R}^{nz}$ are vectors of Lagrange multipliers. Note that all the primal variables and multipliers become implicit functions of η^{mhe} . To simplify the analysis, we suppress this dependence from the notation when the meaning is otherwise clear.

Any solution of a given MHE problem $\mathcal{M}(\eta^{mhe})$ should satisfy the first-order KKT conditions,

$$\nabla_{z_0} \mathcal{L} = \nabla_{z_0} \Gamma + \nabla_{z_0} L_0 - \nabla_{z_0} f_0^T \lambda_1 = 0 \quad (6.7a)$$

$$\nabla_{z_l} \mathcal{L} = \nabla_{z_l} L_l + \lambda_l - \nabla_{z_l} f_l^T \lambda_{l+1} = 0, \quad l = 1, \dots, N-1 \quad (6.7b)$$

$$\nabla_{w_l} \mathcal{L} = \nabla_{w_l} L_l - \nabla_{w_l} f_l^T \lambda_{l+1} = 0, \quad l = 0, \dots, N-1 \quad (6.7c)$$

$$\nabla_{\lambda_{l+1}} \mathcal{L} = z_{l+1} - f_l = 0, \quad l = 0, \dots, N-1 \quad (6.7d)$$

$$\nabla_{z_N} \mathcal{L} = \nabla_{z_N} L_N + \lambda_N = 0 \quad (6.7e)$$

where $f_l := f(z_l, w_l)$, $\chi_l := \chi(z_l)$, $L_l := L(z_l, w_l)$. For later reference we define $A_l = \nabla_{z_l} f_l$, $G_l = \nabla_{w_l} f_l$, $C_l = \nabla_{z_l} \chi_l$. NLP algorithms compute the search step towards the optimal solution by linearizing the nonlinear KKT conditions (6.7) around the current iteration. This gives rise to the KKT system,

$$P_0 \Delta z_0 + F_0 \Delta w_0 - A_0^T \Delta \lambda_1 = -r_{z_0} \quad (6.8a)$$

$$P_l \Delta z_l + F_l \Delta w_l + \Delta \lambda_l - A_l^T \Delta \lambda_{l+1} = -r_{z_l} \quad l = 1, \dots, N-1 \quad (6.8b)$$

$$P_N \Delta z_N + \Delta \lambda_N = -r_{z_N} \quad (6.8c)$$

$$W_l \Delta w_l + F_l^T \Delta z_l - G_l^T \Delta \lambda_{l+1} = -r_{w_l} \quad l = 0, \dots, N-1 \quad (6.8d)$$

$$\Delta z_{l+1} - A_l \Delta z_l - G_l \Delta w_l = -r_{\lambda_{l+1}} \quad l = 0, \dots, N-1. \quad (6.8e)$$

As observed in Chapter 5, if we make an explicit distinction between primal variables and multipliers, the KKT system can be represented as in equation (6.9). Here, $r_{z_l} = \nabla_{z_l} \mathcal{L}$, $r_{w_l} = \nabla_{w_l} \mathcal{L}$, $r_{\lambda_l} = \nabla_{\lambda_l} \mathcal{L}$, $P_l = \nabla_{z_l z_l} \mathcal{L}$, $W_l = \nabla_{w_l w_l} \mathcal{L}$, and $F_l = \nabla_{z_l w_l} \mathcal{L}$, evaluated at the current iteration. In condensed form, this becomes the augmented system (3.6a) solved in IPOPT.

$$\left[\begin{array}{cc|cc} P_0 & F_0 & -A_0^T & \\ F_0^T & W_0 & -G_0^T & \\ & P_1 & F_1 & \\ & F_1^T & W_1 & \\ & & \ddots & \\ & & P_{N-1} & F_{N-1} \\ & & F_{N-1}^T & W_{N-1} \\ & & & P_N \\ \hline -A_0 - G_0 & \mathbb{I}_{n_z} & & \\ & -A_1 - G_1 & \mathbb{I}_{n_z} & \\ & & \ddots & \\ & & & -A_{N-1} - G_{N-1} & \mathbb{I}_{n_z} \end{array} \right] \left[\begin{array}{c} \Delta z_0 \\ \Delta w_0 \\ \Delta z_1 \\ \Delta w_1 \\ \vdots \\ \Delta z_{N-1} \\ \Delta w_{N-1} \\ \Delta z_N \\ \Delta \lambda_1 \\ \vdots \\ \Delta \lambda_{N-1} \\ \Delta \lambda_N \end{array} \right] = - \left[\begin{array}{c} r_{z_0} \\ r_{w_0} \\ r_{z_1} \\ r_{w_1} \\ \vdots \\ r_{z_{N-1}} \\ r_{w_{N-1}} \\ r_{z_N} \\ r_{\lambda_1} \\ \vdots \\ r_{\lambda_{N-1}} \\ r_{\lambda_N} \end{array} \right] \quad (6.9)$$

In the context of MHE, we will denote the KKT matrix as $\mathbf{K}^*(\eta^{mhe}, N)$. As we have seen, we can solve the KKT system efficiently using a direct sparse factorization. Nevertheless, in order to obtain more insight on the particular structure of the MHE problem, we use the forward Riccati decomposition described in Appendix D. This leads to the explicit recursion,

$$\Delta z_N = -\mathbf{\Pi}_N(r_{z_N} + \mathbf{M}_N^{-1}r_{\mathbf{M}_N}) \quad (6.10a)$$

$$\Delta \lambda_l = \mathbf{M}_l^{-1}(\Delta z_l + r_{\mathbf{M}_l}) \quad (6.10b)$$

$$\begin{aligned} \Delta z_{l-1} = & -\mathbf{\Pi}_{l-1}(F_{l-1}W_{l-1}^{-1}G_{l-1}^T - A_{l-1}^T)\Delta \lambda_l \\ & + \mathbf{\Pi}_{l-1}(F_{l-1}W_{l-1}^{-1}r_{w_{l-1}} - r_{z_{l-1}} - \mathbf{M}_{l-1}^{-1}r_{\mathbf{M}_{l-1}}) \end{aligned} \quad (6.10c)$$

$$\Delta w_{l-1} = -W_{l-1}^{-1}F_{l-1}^T\Delta z_{l-1} + W_{l-1}^{-1}G_{l-1}^T\Delta \lambda_l - W_{l-1}^{-1}r_{w_{l-1}} \quad (6.10d)$$

$$l = N, \dots, 1$$

where,

$$\begin{aligned} \mathbf{\Pi}_0 &= (P_0 - F_0W_0^{-1}F_0^T)^{-1} \\ \mathbf{M}_{l+1} &= (G_lW_l^{-1}F_l^T - A_l)\mathbf{\Pi}_l(F_lW_l^{-1}G_l^T - A_l^T) + G_lW_l^{-1}G_l^T \\ \mathbf{\Pi}_{l+1} &= (P_{l+1} + \mathbf{M}_{l+1}^{-1} - F_{l+1}W_{l+1}^{-1}F_{l+1}^T)^{-1} \\ \mathbf{\Pi}_N &= (P_N + \mathbf{M}_N^{-1})^{-1} \\ l &= 0, \dots, N-1 \end{aligned} \quad (6.11)$$

and,

$$\begin{aligned} r_{\mathbf{M}_1} &= r_{\lambda_1} + G_0W_0^{-1}r_{w_0} - (G_0W_0^{-1}F_0^T - A_0)\mathbf{\Pi}_0(r_{z_0} - F_0W_0^{-1}r_{w_0}) \\ r_{\mathbf{M}_{l+1}} &= r_{\lambda_{l+1}} + G_lW_l^{-1}r_{w_l} - (G_lW_l^{-1}F_l^T - A_l)\mathbf{\Pi}_l(r_{z_l} + \mathbf{M}_l^{-1}r_{\mathbf{M}_l} - F_lW_l^{-1}r_{w_l}) \\ l &= 0, \dots, N-1 \end{aligned} \quad (6.12)$$

As with NMPC, note that the computational complexity of the MHE Riccati decomposition strategy scales as $O(N(n_z + n_w)^3)$. This is an important observation because a similar

strategy is used in the Kalman Filter [21]. For systems with a large number of states and disturbances, this is not a computationally efficient approach. Moreover, in the presence of unstable dynamics and/or non-informative measurements, the recursion can also lead to numerical ill-conditioning. The ability to handle these degeneracies directly through the NLP solver (e.g. using an inertia correction strategy), is one of the main practical advantages of using a direct factorization strategy. Finally, note that the inertia correction strategy of IPOPT can also be used in the context of state estimation to infer the observability properties of the dynamic system.

6.2.2 Covariance Information

Extracting covariance information from the solution of the MHE problem is important since it can be used as a measure of uncertainty and to update the arrival cost [102]. In particular, note that setting $\bar{\Pi}_0(k+1) = \mathbf{M}_1$ with \mathbf{M}_1 extracted from (6.11) provides a stronger update than the traditional EKF update in (6.5) since it incorporates exact second order information of the model. To see this, we extract \mathbf{M}_1 from (6.11),

$$\begin{aligned}\mathbf{\Pi}_0 &= (P_0 - F_0 W_0^{-1} F_0^T)^{-1} \\ \mathbf{M}_1 &= (G_0 W_0^{-1} F_0^T - A_0) \mathbf{\Pi}_0 (F_0 W_0^{-1} G_0^T - A_0^T) + G_0 W_0^{-1} G_0^T.\end{aligned}\quad (6.13)$$

If we neglect all the second order contributions of the model and the cross interaction term F_0 then from (6.1) and (6.2), $P_0 = \bar{\Pi}_0(k)^{-1} + C_0^T R_0^{-1} C_0$ and $W_0^{-1} = Q_0$. If we apply the matrix inversion lemma to P_0^{-1} we obtain,

$$\begin{aligned}\mathbf{\Pi}_0 &= \bar{\Pi}_0(k) - \bar{\Pi}_0(k) C_0^T (R_0 + C_0 \bar{\Pi}_0(k) C_0^T)^{-1} C_0 \bar{\Pi}_0(k) \\ \mathbf{M}_1 &= A_0 \mathbf{\Pi}_0 A_0^T + G_0 Q_0 G_0^T\end{aligned}\quad (6.14)$$

which is the simplified EKF update formula (6.5). In classical estimation literature, the simplified matrices \mathbf{M}_l , $l = 1, \dots, N$ are known as the *prior* covariance matrices [21]. Using similar simplifications, we can show that matrices $\mathbf{\Pi}_l$, $l = 1, \dots, N$ in (6.11) become,

$$\mathbf{\Pi}_l^{-1} = \mathbf{M}_l^{-1} + C_l^T R_l^{-1} C_l, \quad l = 1, \dots, N. \quad (6.15)$$

These are normally known as the *posterior* covariance matrices where $\mathbf{\Pi}_N$ is the covariance matrix of the current state estimate. The exact representation of this matrix can be obtained from recursion (6.11).

If a Riccati decomposition is used to solve the KKT system in a tailored NLP solver, the covariance matrices $\mathbf{\Pi}_N$ and \mathbf{M}_1 are obtained as a natural outcome of the recursion. However, if a direct factorization is used (as in a general NLP solver like IPOPT) these matrices are never formed. To avoid this limitation, note from (6.10a) that if we set $r_{z_N} = -\mathbb{I}_{n_z}(:, j)$ in (6.9) at the solution (rest of the right-hand sides are zero) where $\mathbb{I}_{n_z}(:, j)$ is the j -th column of the identity matrix then, $\Delta z_N = \mathbf{\Pi}_N(:, j)$. This can be illustrated from the KKT system (6.9),

$$\left[\begin{array}{cc|cc} P_0 & F_0 & -A_0^T & -G_0^T \\ F_0^T & W_0 & \mathbb{I}_{n_z} & -A_1^T \\ & P_1 & F_1 & -G_1^T \\ & F_1^T & W_1 & \\ & & \ddots & \\ & & P_{N-1} & F_{N-1} \\ & & F_{N-1}^T & W_{N-1} \\ & & & P_N \\ \hline -A_0 - G_0 \mathbb{I}_{n_z} & & & \mathbb{I}_{n_z} \\ & -A_1 - G_1 \mathbb{I}_{n_z} & & \\ & & \ddots & \\ & & & -A_{N-1} - G_{N-1} \mathbb{I}_{n_z} \end{array} \right] \begin{bmatrix} \Delta z_0 \\ \Delta w_0 \\ \Delta z_1 \\ \Delta w_1 \\ \vdots \\ \Delta z_{N-1} \\ \Delta w_{N-1} \\ \mathbf{\Pi}_N(:, j) \\ \Delta \lambda_1 \\ \vdots \\ \Delta \lambda_{N-1} \\ \Delta \lambda_N \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \mathbb{I}_{n_z}(:, j) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (6.16)$$

This implies that the covariance matrix can be computed efficiently by performing n_z backsolves with the factored KKT matrix at the solution. This is an interesting result because this implies that the reduced Hessian extraction capability described in Chapter 3 can also be used in this context.

Extracting \mathbf{M}_1 from the KKT matrix is not as straightforward. From (6.9) and (6.10) we can prove that, setting all the right-hand sides in (6.9) to zero, adding the constraint $\Delta z_1 = -\mathbb{I}_{n_z}(:, j)$ and dropping the corresponding rows in the KKT system by adding slack variables $\Delta \nu_u$ recursively, leads to $\Delta \lambda_1 = \mathbf{M}_1^{-1}(:, j)$. That is,

3. At the solution $s^*(\eta_0^{mhe}, N+1)$, hold factors of KKT matrix $\mathbf{K}^*(\eta_0^{mhe}, N+1)$ or compute sensitivity matrix $\frac{\partial s^*}{\partial \eta^{mhe}}$.

On-line, at t_{k+1} :

1. Obtain the true measurement $y(k+1)$ and define the *true* problem data $\eta^{mhe} = \eta^{mhe}(k+1)$. Compute an instantaneous approximate solution $\tilde{s}(\eta^{mhe}, N+1)$ from sensitivity (3.33) or as a perturbed Newton step (3.36) and extract $\tilde{x}^{as}(k+1) = \tilde{z}_{N+1}$ and $\bar{z}_0(k+1) = \tilde{z}_1$. Extract $\bar{\Pi}_0(k+1) = \mathbf{M}_1$ from the fixed KKT matrix $\mathbf{K}^*(\eta_0^{mhe}, N+1)$.
2. Update data, set $k := k+1$ and return to the background process.

If the above shifting strategy is used, the NLP sensitivity perturbation is $|\eta^{mhe} - \eta_0^{mhe}| = |y(k+1) - \bar{y}(k+1)|$ so that the approximation error between the approximate and the ideal state estimates is,

$$|\tilde{x}^{as}(k+1) - \tilde{x}^{id}(k+1)| = O(|y(k+1) - \bar{y}(k+1)|^2). \quad (6.18)$$

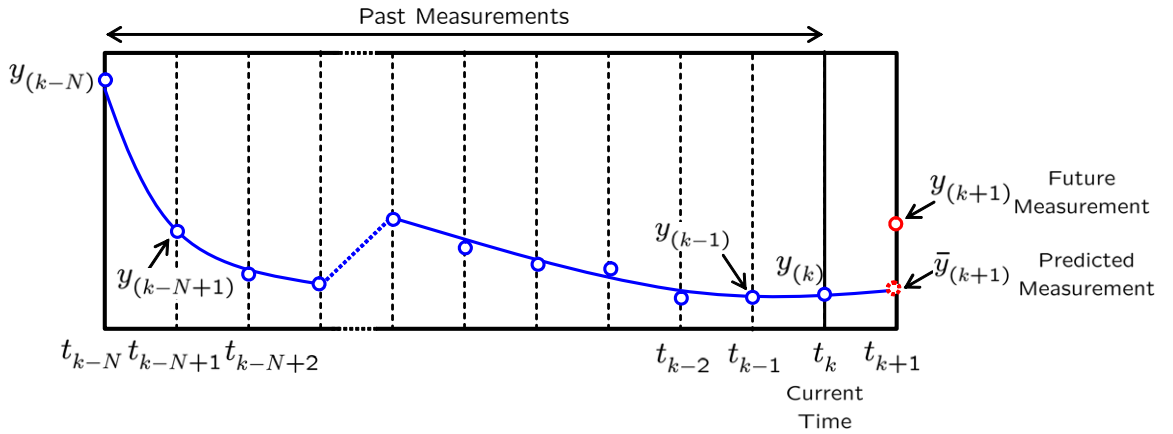


Figure 6.1: Schematic representation of estimation horizon for extended MHE problem.

Note that the covariance matrix $\bar{\Pi}_0(k+1)$ is extracted from $\mathbf{K}^*(\eta_0^{mhe}, N+1)$ which is evaluated at $s^*(\eta_0^{mhe}, N+1)$. The optimal covariance matrix should be extracted from $\mathbf{K}^*(\eta^{mhe}, N+1)$ evaluated at $s^*(\eta^{mhe}, N+1)$. From Lipschitz continuity of the KKT matrix

(implied in Theorem 3.2.2) it can be shown that $|\mathbf{K}^*(\eta^{mhe}, N+1) - \mathbf{K}^*(\eta_0^{mhe}, N+1)| \leq L_s |\eta^{mhe} - \eta_0^{mhe}|$ for a given positive constant L_s . That is, the error bound between the approximate and optimal covariance matrices can be shown to be $O(|y(k+1) - \bar{y}(k+1)|)$.

From the optimality conditions (6.7) of the extended problem $\mathcal{M}_{N+1}(\bar{\eta}^{mhe}(k+1))$ we can show that $\lambda_{N+1}^* = 0$ since, by construction, $\bar{y}(k+1) = \chi_{N+1}(z_{N+1})$. In addition, this implies that $w_N^* = 0$. At this nominal solution, the last element of the recursion (6.10) becomes,

$$\begin{aligned} \Delta z_{N+1} &= -\mathbf{\Pi}_{N+1} r_{z_{N+1}} \\ &= \mathbf{\Pi}_{N+1} C_{N+1}^T R_{N+1}^{-1} (\bar{y}(k+1) - \chi_{N+1}(z_{N+1}^*(\bar{y}(k+1)))) . \end{aligned} \quad (6.19)$$

Once the new measurement $y(k+1)$ is obtained, we perturb the right hand side so that,

$$\begin{aligned} \Delta z_{N+1} &= \tilde{z}_{N+1}(y(k+1)) - z_{N+1}^*(\bar{y}(k+1)) \\ &= \mathbf{\Pi}_{N+1} C_{N+1}^T R_{N+1}^{-1} (y(k+1) - \chi_{N+1}(z_{N+1}^*(\bar{y}(k+1)))) \\ &= K_{N+1} (y(k+1) - \chi_{N+1}(z_{N+1}^*(\bar{y}(k+1)))) . \end{aligned} \quad (6.20)$$

Note that $\bar{y}(k+1) = \chi_{N+1}(z_{N+1}^*(\bar{y}(k+1)))$ and $\tilde{x}^{as}(k+1) = \tilde{z}_{N+1}(y(k+1))$ so that,

$$\tilde{x}^{as}(k+1) = z_{N+1}^*(\bar{y}(k+1)) + K_{N+1} (y(k+1) - \bar{y}(k+1)) . \quad (6.21)$$

This expression can be seen as an analog of the Kalman Filter update formula where K_{N+1} is the so-called Kalman matrix [21]. Here, $z_{N+1}^*(\bar{y}(k+1))$ can be interpreted as the *before-measurement* estimate. The sensitivity estimate $\tilde{x}^{as}(k+1) = \tilde{z}_{N+1}(y(k+1))$ can be interpreted as the *after-measurement* estimate which is an approximation of the optimal estimate $\tilde{x}^{id}(k+1) = z_{N+1}^*(y(k+1))$.

The above expression can also be used to analyze the impact of the characteristics of the dynamic system and of the design of the MHE problem on the approximation error. For instance, it is clear that as $N \rightarrow \infty$ (i.e. we add more information) then $\mathbf{\Pi}_{N+1} \rightarrow \mathbf{\Pi}_\infty$ (the steady-state covariance matrix) [102]. Therefore, the impact of the update becomes

constant and small and the approximation error $|\tilde{x}^{as}(k+1) - \tilde{x}^{id}(k+1)| \rightarrow \delta_\infty$ for $\delta_\infty > 0$. Similarly, if the system is strongly observable then the eigenvalues of $\mathbf{\Pi}_{N+1}$ will be positive and small and the approximation error will also tend to be small (i.e. the impact of an additional measurement on the existent sequence is not that strong).

Remark: The proposed asMHE algorithm assumes that the background problem \mathcal{M}_{N+1} can be solved within one sampling time. As with NMPC, we can warm-start the background problems by shifting the NLP sensitivity solution one step forward to reduce the number of iterations.

Remark: The solution of an extended problem with $N+1$ steps is proposed in order to minimize the approximation error and to achieve *consistency* with the ideal MHE estimator (7.3) which has well-established convergence and stability properties [102]. For instance, note that in order to solve the ideal MHE problem $\mathcal{M}_N(\eta^{mhe}(k+1))$ at t_{k+1} we need the *a priori* estimate $z_{1|k}^*$ and covariance \mathbf{M}_1 extracted from the solution of $\mathcal{M}_N(\eta^{mhe}(k))$. Note also that the solution of the predicted problem $\mathcal{M}_{N+1}(\bar{\eta}^{mhe}(k+1))$ for $l = 0, \dots, N$ and the solution of $\mathcal{M}_N(\eta^{mhe}(k))$ are *exactly the same* since the predicted measurement $\bar{y}(k+1)$ *does not add information into the problem*. As a consequence, the solution of the extended problem implicitly gives us the required *prior* information. Nevertheless, if the prior covariance matrix does not need to be updated at each time step, it is possible to solve the predicted problem with N steps.

6.4 Illustrative Example

We consider a simulated MHE scenario on the nonlinear continuous stirred tank reactor described in Appendix C. The objective of this case study is to analyze the effect of NLP sensitivity errors on the performance of the asMHE estimator and contrast this with the performance of the optimal or ideal MHE estimator. We use batch data generated from a simulated scenario. The continuous-time dynamic model is converted to the discrete-time

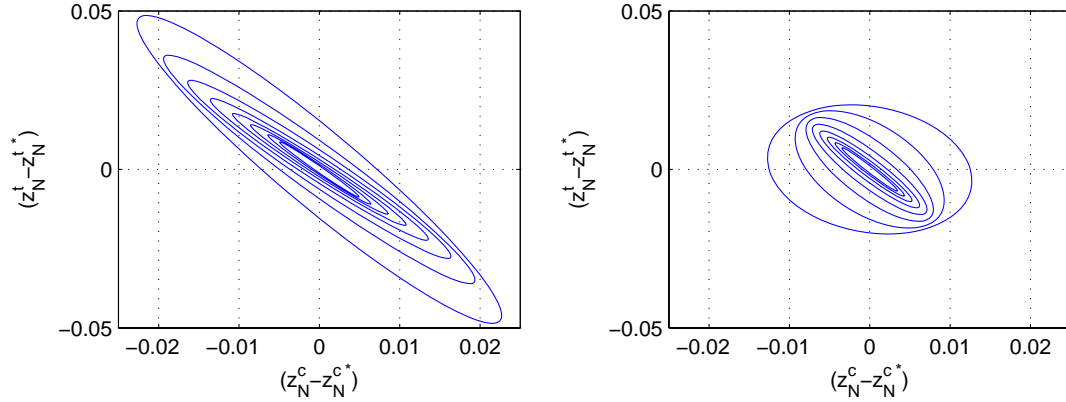


Figure 6.2: Effect of estimation horizon length N on the states covariance. Concentration z_c is measurement (left). Temperature z_t is measurement (right).

form using orthogonal collocation on finite elements. The finite elements are placed in order to match the sampling times. The resulting NLPs can be solved with IPOPT in a negligible amount of time. In Chapter 8 we will see that the performance characteristics apply to much larger examples as well. We use the reduced Hessian extraction and NLP sensitivity capabilities described in Chapter 3.

We first compare the effect of the estimation horizon length N on the covariance matrix $\mathbf{\Pi}_N$ of the state estimates for two cases. In the first case we assume that the concentration z^c is used as measurement to infer the temperature z^t . In the second case we use z^t to infer z^c . The shortest estimation horizon used contains 5 time steps (outer ellipsoids) while the longest contains 50 time steps (inner ellipsoids). The resulting 95% confidence regions are presented in Figure 6.2. It is clear that the eigenvalues of the covariance matrices decrease as the horizon is increased. Moreover, note that the ellipsoids of the left graph (concentration measured) are relatively larger than those on the right graph (temperature measured). Consequently, we can conclude that the temperature z^t is the most informative measurement.

Scenario 1. The simulated states are corrupted with Gaussian noise ($\sigma = 0.01$) and we use $\bar{z}_0 = [0.3 \ 0.3]$ and $\bar{\Pi}_0^{-1} = \text{diag}\{0.1, 0.1\}$ as initial guess. The weights for the output deviations are set to $\frac{1}{\sigma^2}$. In Figure 6.3 we compare the performance of both the ideal and asMHE algorithms. As can be seen from subplot b), both algorithms are able to reconstruct the true state z^c of the system based on temperature information. The asMHE estimator is able to remove computational delay. Performance deterioration of asMHE due to NLP sensitivity errors is not immediately evident from these profiles but can be appreciated by comparing the cost functions of both MHE algorithms in subplot c). It is interesting to observe that the performance of the asMHE algorithm degrades at some time steps creating small deviations from the optimal cost function of the ideal MHE. This is due to the fact that the difference between the predicted and actual measurement is quite large, leading to large approximation errors. The difference in performance tends to disappear as information is accumulated and identical performance is observed for the rest of the operating horizon even in the presence of large levels of noise. These results are in agreement with equation (6.21) and the observations of the previous sections.

Scenario 2. We add two disturbances on the activation energy E_a to test the robustness of both MHE estimators. As can be seen from subplots a) and b) in Figure 6.4, the disturbance jumps at time step 75 and 100 disrupt the inferred state profile, but the estimators are able to reject the disturbances. Interestingly, from the cost functions of subplot c) we see that the approximation errors do not degrade the performance of the asMHE estimator.

6.5 Concluding Remarks

An advanced-step moving horizon estimation (MHE) algorithm is presented in this chapter. The strategy solves a reference problem in between sampling times using a predicted future measurement, and corrects on-line using fast nonlinear programming sensitivity calculations. Rigorous performance bounds are derived based on classical NLP sensitivity results.

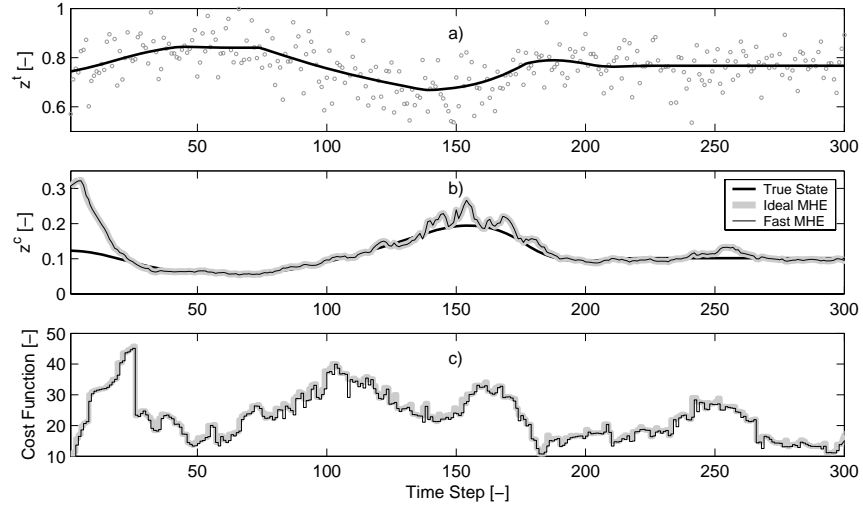


Figure 6.3: Performance comparison of advanced-step and ideal MHE strategies in the presence of measurement noise.

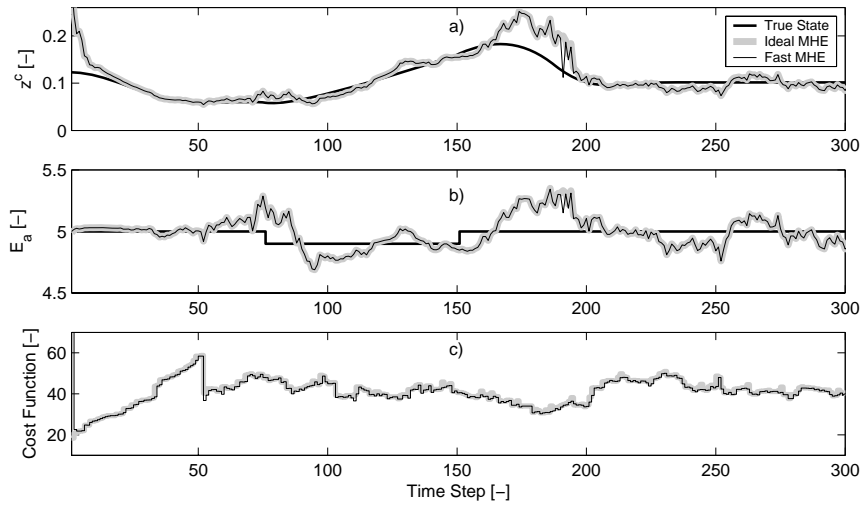


Figure 6.4: Performance comparison of advanced-step and ideal MHE strategies in the presence of noise and disturbances.

A detailed analysis of the KKT conditions of the MHE problem is performed in order to derive strategies for covariance information extraction from general NLP solvers and to establish connections with the traditional Kalman filter. It is demonstrated through a small simulation study that the proposed algorithm is able to mimic the optimal performance of ideal MHE. Having derived advanced-step formulations for NMPC and MHE, in the following chapter we study the output feedback problem in which we couple them.

Chapter 7

NMPC-MHE Coupling

In this chapter, we discuss issues related to the coupling of advanced-step strategies for NMPC and MHE. In particular, we will demonstrate that the advanced-step NMPC controller remains stable in the face of sensitivity approximation errors arising from both the estimator and the controller.

7.1 Standard NMPC-MHE Formulation

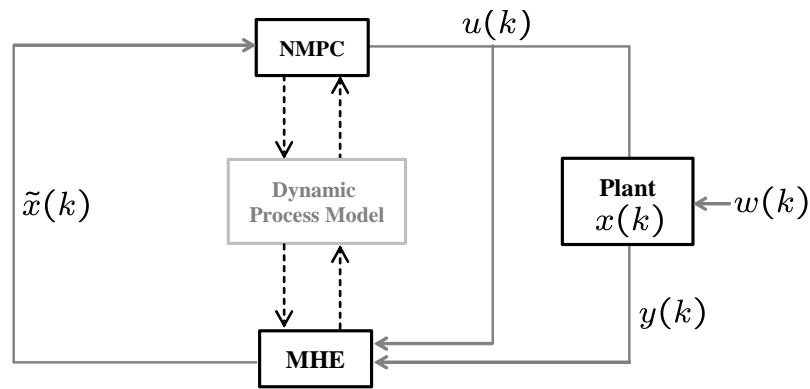


Figure 7.1: Scenario arising in NMPC and MHE coupling.

We make use of Figure 7.1 to describe a typical scenario arising in the interaction between MHE and NMPC. To start the discussion, we consider a discrete-time dynamic model of

an *uncertain* plant of the form,

$$x(k+1) = \hat{f}(x(k), u(k), w(k)) = f(x(k), u(k)) + g(x(k), u(k), w(k)) \quad (7.1a)$$

$$y(k+1) = \chi(x(k+1)) + v(k+1), \quad (7.1b)$$

where $x(k)$ is the *true* plant state at time instant t_k and $u(k)$ is the implemented control action. The observed output $y(k)$. The true plant deviates from the nominal prediction due to the process disturbance $w(k)$ and measurement noise $v(k+1)$.

Assume that the plant is currently located at sampling time t_k with the output and input measurements $y(k-N), \dots, y(k)$ and $u(k-N), \dots, u(k-1)$, respectively. In addition, we have an *a priori* estimate of the past state of the plant $\bar{z}_0(k)$ with covariance $\bar{\Pi}_0^{-1}(k)$ that generated this measurement history. For this analysis, we will define the MHE problem data¹ as,

$$\eta^{mhe}(k) := \{y(k-N), \dots, y(k)\}. \quad (7.2)$$

Using this information, we would like to compute an estimate $\tilde{x}^{id}(k)$ of the current state $x(k)$. In order to do this, we solve the MHE problem,

$$\begin{aligned} \mathcal{M}_N(\eta^{mhe}(k)) \quad \min_{z_0, w_l} \quad & \phi(\eta^{mhe}(k)) = \Gamma(z_0) + L_N(z_N) + \sum_{l=0}^{N-1} L_l(z_l, w_l) \\ \text{s.t.} \quad & z_{l+1} = \hat{f}(z_l, w_l), \quad l = 0, \dots, N-1 \\ & z_l \in \mathbb{Z}, \quad w_l \in \mathbb{W}. \end{aligned} \quad (7.3)$$

From the solution of this problem, we obtain the optimal disturbance sequence $\{w_0^* \dots w_{N-1}^*\}$ and state sequence $\{z_0^* \dots z_N^*\}$ from which we extract the current state estimate of the plant $\tilde{x}^{id}(k) = z_N^*$ with associated estimation error $e(k) := \tilde{x}^{id}(k) - x(k)$. Using this estimate, we

¹The *prior* information and the input measurements are not defined here as part of the problem data. This can be done without loss of generality since, in the advanced-step MHE strategy presented in Chapter 6, the sensitivity perturbation is only introduced through the output measurements.

define the NMPC problem data as $\eta^{mpc}(k) := \tilde{x}^{id}(k)$. The NMPC problem is,

$$\begin{aligned} \mathcal{P}_N(\eta^{mpc}(k)) \quad \min_{v_l, z_l} \quad J_N(\eta^{mpc}(k)) &:= F(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \\ \text{s. t. } z_{l+1} &= f(z_l, v_l), \quad z_0 = \tilde{x}^{id}(k) \quad l = 0, \dots, N-1 \\ z_l &\in \mathbb{X}, \quad z_N \in \mathbb{X}_f, \quad v_l \in \mathbb{U}. \end{aligned} \quad (7.4)$$

The control action is extracted from the optimal trajectory $\{z_0^* \dots z_N^*, v_0^*, \dots, v_{N-1}^*\}$ as $u(k) = v_0^* := h^{id}(\tilde{x}^{id}(k))$. Here, $h^{id}(\cdot)$ denotes the ideal NMPC feedback law. Note that this control action is *imperfect* since the true state of the plant is $x(k)$ and not the estimate $\tilde{x}^{id}(k)$. That is, the estimation error acts as an additional disturbance. At the next time step t_{k+1} , the plant will evolve as,

$$x(k+1) = \hat{f}(x(k), h^{id}(\tilde{x}^{id}(k)), w(k)), \quad (7.5a)$$

$$y(k+1) = \chi(x(k+1)) + v(k+1). \quad (7.5b)$$

We shift the measurement sequence one step forward and update the prior information (if necessary). We define the new data vector,

$$\eta^{mhe}(k+1) := \{y(k-N+1), \dots, y(k+1)\} \quad (7.6)$$

and we solve the new MHE problem. Having the new state estimate $\tilde{x}^{id}(k+1)$ we solve the next NMPC problem.

Since feedback is provided through the state estimate inferred from the output measurements and not from the true state, this problem is normally known as the output feedback problem. In general, the stability of this control scheme can only be guaranteed for a stable estimator (i.e. stable or decaying estimator error) and by making use of the inherent robustness properties of NMPC [90]. In addition, note that the NMPC-MHE coupling introduces severe computational limitations since the feedback delay is now equal to the solution time of the MHE problem plus the solution time of the NMPC problem.

7.2 Advanced-Step Formulation

It is possible to minimize the on-line time required to solve the MHE problem and then the NMPC problem to two fast backsolves using an advanced-step framework [126, 133]. Imagine that at time t_k we know the control action $u(k)$ and we would like to obtain an estimate of the future state $x(k+1)$ but we don't know the future measurement $y(k+1)$. Nevertheless, we can use the current estimate $\tilde{x}^{id}(k)$ and control $u(k)$ to predict the future state and associated measurement,

$$z(k+1) = f(\tilde{x}^{id}(k), u(k)), \quad (7.7a)$$

$$\bar{y}(k+1) = \chi(z(k+1)), \quad (7.7b)$$

and use this to complete the problem data,

$$\bar{\eta}^{mhe}(k+1) := \{y(k-N), \dots, \bar{y}(k+1)\} \quad (7.8)$$

to start the solution of the predicted MHE problem $\mathcal{M}_{N+1}(\bar{\eta}^{mhe}(k+1))$ with $N+1$ time steps. We use the predicted state to define $\bar{\eta}^{mpc}(k+1) := z(k+1)$ and start the solution of the predicted NMPC problem $\mathcal{P}_N(\bar{\eta}^{mpc}(k+1))$. Note that both problems are *decoupled* so this can be done simultaneously and thus reduce the sampling time [43]. At the solution of these problems, we hold the corresponding KKT matrices \mathbf{K}_*^{mhe} and \mathbf{K}_*^{mpc} of the MHE and NMPC problems, respectively.

Once the true measurement $y(k+1)$ becomes available, we compute a fast backsolve with \mathbf{K}_*^{mhe} to obtain the sensitivity approximation \tilde{s}^{mhe} from which we extract the *approximate* state estimate $\tilde{x}^{as}(k+1)$. This estimate differs from the optimal state estimate $\tilde{x}^{id}(k+1)$ and the true state $x(k+1)$. Using the approximate state estimate we perform a fast backsolve with \mathbf{K}_*^{mpc} to obtain the sensitivity approximation \tilde{s}^{mpc} from which we extract the approximate control action,

$$u(k+1) = h^{as}(\tilde{x}^{as}(k+1)) \quad (7.9)$$

which differs from the optimal control $h^{id}(\tilde{x}^{id}(k+1))$.

To warm-start the background problems at the next sampling time, we use the approximate sensitivity solutions \tilde{s}^{mhe} and \tilde{s}^{mpc} to generate the *shifted* warm-start sequences for the next problems $\mathcal{M}_{N+1}(\bar{\eta}^{mhe}(k+2))$ and $\mathcal{P}_N(\bar{\eta}^{mpc}(k+2))$ [43]. From the solution of these problems, we update the KKT matrices in background. Note that the approximate solutions \tilde{s}^{mhe} and \tilde{s}^{mpc} can also be refined in background using fixed-point iterations (equation (3.37)) with \mathbf{K}_*^{mhe} and \mathbf{K}_*^{mpc} *before* using them to generate the warm-start sequences. We can summarize the proposed framework as follows:

In background, between t_k and t_{k+1} :

1. Use the *current* estimate $\tilde{x}^{as}(k)$ and the control $u(k)$ to predict the future state $z(k+1) = f(\tilde{x}^{as}(k), u(k))$ and corresponding output measurement $\bar{y}(k+1) = \chi(z(k+1))$.

2. Define the data

$$\bar{\eta}^{mhe}(k+1) = \{y(k-N), \dots, \bar{y}(k+1)\}$$

and $\bar{\eta}^{mpc}(k+1) = z(k+1)$. Use the available warm-start points s_o^{mhe} and s_o^{mpc} to solve the predicted problems $\mathcal{M}_{N+1}(\bar{\eta}^{mhe}(k+1))$ and $\mathcal{P}_N(\bar{\eta}^{mpc}(k+1))$.

3. Hold the KKT matrices \mathbf{K}_*^{mhe} and \mathbf{K}_*^{mpc} .

On-line, at t_{k+1} :

1. Obtain the true measurement $y(k+1)$ and define the *true* MHE data $\eta^{mhe}(k+1)$. Reuse factorization of \mathbf{K}_*^{mhe} to compute a fast approximation \tilde{s}^{mhe} from (3.36) and extract $\tilde{x}^{as}(k+1)$.
2. Use $\tilde{x}^{as}(k+1)$ to define the *true* NMPC problem data $\eta^{mpc}(k+1)$. Reuse the factorization of \mathbf{K}_*^{mpc} to compute fast approximation \tilde{s}^{mpc} from (3.36) and extract $u(k+1) = h^{as}(\tilde{x}^{as}(k+1))$.

3. If necessary, refine \tilde{s}^{mhe} and \tilde{s}^{mpc} . Generate the warm-starts s_o^{mhe} and s_o^{mpc} , set $k := k + 1$ and return to background.

7.2.1 Stability Issues

It is clear that both the state estimate and the associated control action are *suboptimal* due to the presence of NLP approximation errors. In particular, note that the approximation error of the asMHE estimator propagates to the sensitivity error of the controller counterpart. Here, we are interested in assessing the impact of these accumulated errors in the stability of the output feedback system. From the controller point of view, we are interested in finding sufficient conditions under which the closed-loop remains stable in the face of disturbances and NLP sensitivity errors.

For the optimal or ideal NMPC controller (instantaneous optimal solutions), we consider the neighboring costs of the extended problems (5.22) with *perfect* state information $J_{x(k)}^{hid(x(k))} := J_{N+1}(x(k), h^{id}(x(k)))$ and $J_{x(k+1)}^{hid(x(k+1))} := J_{N+1}(x(k+1), h^{id}(x(k+1)))$ as reference points. As observed by Muske and Rawlings [90], since the *implemented* control action is based on the state *estimate* $\tilde{x}^{id}(k)$ coming from MHE and not on the true state $x(k)$, we consider this as an additional disturbance to the closed-loop system through the cost $J_{\hat{x}(k+1)}^{hid(\hat{x}(k+1))}$ where $\hat{x}(k+1) = \hat{f}(x(k), h^{id}(\tilde{x}^{id}(k)), w(k))$. From Lipschitz continuity of the cost function we have,

$$|J_{\hat{x}(k+1)}^{hid(\hat{x}(k+1))} - J_{x(k+1)}^{hid(x(k+1))}| \leq L_J L_f L_h \|x(k) - \tilde{x}^{id}(k)\|.$$

Explicit bounds and convergence properties on the estimator error $\|x(k) - \tilde{x}^{id}(k)\|$ can be established for the MHE formulation (7.3) and related variants [102, 3]. With this, we can treat the estimator error as a bounded disturbance $g(x(k), u(k), w(k))$ and define $\tilde{x}^{id}(k) := x(k) + g(x(k), u(k), w(k))$. This allows us to restate the following robustness result for the combined *asMHE* and *asNMPC* strategies.

Theorem 7.2.1 (*Theorem 5.3.2*). *Assume that the NLPs for (7.3) and (7.4) can be solved within one sampling time, and that nominal and robust stability assumptions (Assumption 1) for ideal NMPC hold, then there exist bounds on the noise w and v for which the cost function $J_{N+1}(x)$, obtained from the combined asMHE-asNMPC strategy, is an ISS-Lyapunov function and the resulting closed-loop system is ISS stable.*

7.3 Illustrative Example

We analyze the impact of increasingly stronger perturbations on the performance of the advanced-step MHE/NMPC controller. The system is the nonlinear CSTR reactor described in Appendix C. In this scenario, it is assumed that the temperature z^t is *measured* and the concentration z^c is *inferred*. The objective of the controller is to manipulate the cooling water flow rate u to perform two transitions while rejecting different levels of Gaussian noise with standard deviation σ . The noise is measured as a percentage deviation from the nominal measurement value. The noise is added at each time step to the controller predicted temperature to generate the plant data response $x(k)$ and associated measurement $y(k)$. As we increase the noise level, the NLP sensitivity approximations of both the asMHE estimator and asNMPC controller become less accurate. The estimator error is initialized at zero. Therefore, the measurement noise is the only disturbance arising in the system.

As can be seen in Figure 7.2, if no measurement noise is added, the performance of the approximate asMHE/NMPC controller is identical to that of an ideal counterpart. Since the estimation error is zero at the beginning of the horizon, the predicted and actual measurements are the same at each time step. Consequently, no NLP sensitivity errors are introduced.

From Figure 7.3 we can see that the asMHE/NMPC controller is able to remove the feedback delay and stabilize the system around the unstable points in the presence of a medium level of noise of $\sigma = 5\%$. Note that the random perturbations drive the closed-loop

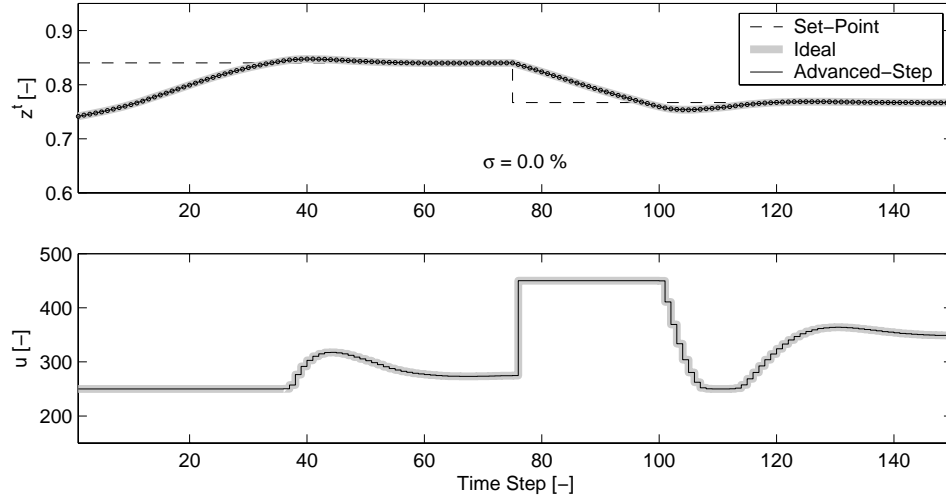


Figure 7.2: Performance of asMHE/NMPC controller in the absence of measurement noise.

response to a completely different path compared to the previous case. The performance is slightly affected by the sensitivity approximation errors. In particular, it is interesting to observe that the control actions of the asNMPC and ideal controller are nearly identical. This indicates that the asMHE estimator acts as a filter or *buffer* to the controller.

In Figure 7.4 we note that for a large level of noise of $\sigma = 7.5\%$, the approximation errors for both the estimator and the controller become larger but the closed-loop system remains stable. The difference becomes evident when we compare the control actions of the approximate and ideal controllers during the *transitions*. Nevertheless, note that when the system stabilizes and the estimator error converges to zero both responses become identical.

7.4 Concluding Remarks

In this chapter, we analyze issues related to the coupling of advanced-step MHE and NMPC strategies. Here, we see that the advanced-step strategy allows to decouple the background MHE and NMPC problems and minimizes the on-line solution time to two fast backsolves.

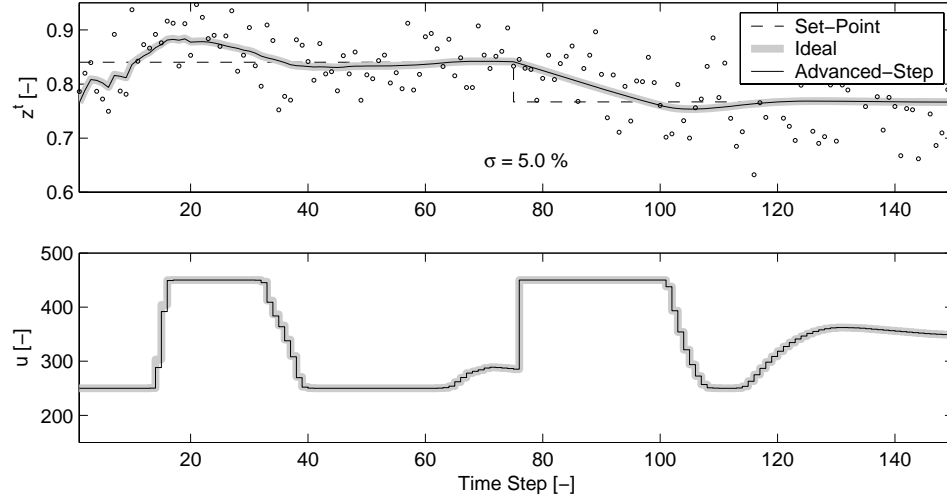


Figure 7.3: Performance of asMHE/NMPC controller with medium level of measurement noise $\sigma = 5\%$.

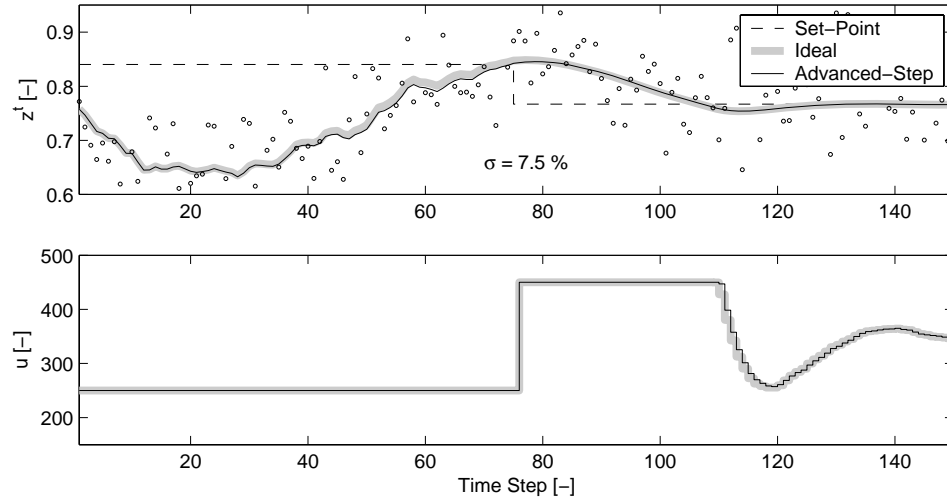


Figure 7.4: Performance of asMHE/NMPC controller with large level of measurement noise $\sigma = 7.5\%$.

We briefly discuss stability issues to argue that stability bounds for the advanced-step NMPC controller can be established in the presence of a stable MHE estimator. We demonstrate through a small case study that the proposed approximate strategies mimic the performance of an ideal output feedback NMPC-MHE controller. In the following chapter we will demonstrate that these developments are applicable to large-scale systems as well.

Part III

Industrial Application

Chapter 8

Industrial Low-Density Polyethylene Case Study

In this chapter, we make use of the proposed computational strategies to develop a model-based operations framework for an industrial low-density polyethylene (LDPE) tubular reactor process. We demonstrate that the strategies can handle a highly sophisticated first-principles model and this can be exploited to improve the profitability of the process.

8.1 Introduction

Low-density polyethylene (LDPE) is an important commodity polymer in today's economy due to its high flexibility and relatively low-cost [75]. LDPE is mostly produced in tubular reactors by free-radical polymerization of ethylene at supercritical conditions (2000-3000 atm and 150-350°C). A typical tubular reactor and corresponding temperature profiles for the reactor core and jackets are sketched in Figure 8.1. LDPE reactors consist of long pipes (1-3 km) with small inner diameters (5-10 cm) and thick reactor walls (2-5 cm) which are divided into several reaction and cooling zones. Each zone is equipped with a jacket cooling system used to remove the large amounts of heat produced by polymerization. Multiple side streams containing monomer, comonomer, chain transfer agent (CTA), and initiators can be fed along the reactor to control the temperature profile and the resulting polymer properties. The large heat transfer areas and low degrees of back-mixing resulting in these units permit the high throughput production of LDPE resins with unique processability

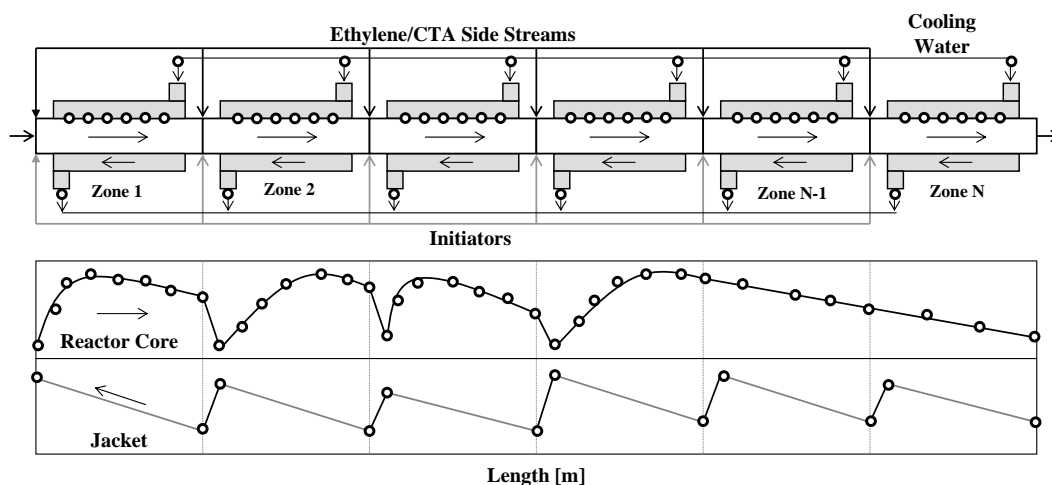


Figure 8.1: Schematic representation of multi-zone LDPE tubular reactor (top). Typical reactor core and jacket temperature profiles (bottom).

and end-use properties.

Despite the multiple benefits offered by LDPE tubular reactors, there exist several factors limiting their performance. The first issue arises due to their distributed and multivariable nature which gives rise to complex interactions along the pipe. This significantly complicates the operability of these processes. The most common approach to cope with this complexity is to find operating conditions able to produce a particular grade by trial and error and/or experience. The resulting recipes are enforced *strictly* through an appropriate regulatory control system. While these recipes work well in many cases, they tend to be rather conservative and need to be constantly adapted for each new grade incorporated into the product portfolio. A second important problem arising in LDPE reactors is the persistent and slow deposition of polymer on the inner reactor walls [76, 22, 23]. The resulting fouling layer is highly insulating and decreases the heat-transfer rate to the cooling jacket. Since the polymerization reactions are highly exothermic, the production rate needs to be dropped progressively in order to keep the temperature profile within safe limits and avoid thermal runaway. The impact of fouling on the overall profitability of high-throughput

LDPE reactors is extremely large. For instance, a decrease of 1% in the daily average production rate can easily translate into millions of dollars of annual profit losses.

The potential economic benefits and high operational complexity of LDPE reactors have motivated research efforts in many areas. Extensive experimental studies have been performed in order to understand the fundamental interactions between the reactor design and operating conditions and the resulting polymer properties [84, 51, 70]. This increased level of understanding has translated into numerous first-principles models of different complexity [73, 72, 124, 20, 19]. Some of these models have been used for off-line tasks such as reactor design and dynamic transient analysis [97, 56]. As a natural step, it is desired to use these models to perform on-line tasks such as real-time optimization and model-based control [74, 127]. However, this has been so hindered by the high computational complexity associated with LDPE tubular reactor models. In this chapter, we make use of the previously proposed computational strategies to perform different optimization-based tasks arising in the development of a model-based operations framework for LDPE processes.

8.2 Mathematical Modeling and Parameter Estimation

A number of comprehensive LDPE tubular reactor models are available in the literature [51, 20, 56]. These models differ in the mechanisms postulated to describe the polymerization kinetics [70], the prediction approach of the final polymer properties [1], the prediction methods of the reacting mixture physical properties [19], assumptions regarding the flow regime and dynamic responses [56], and, finally, in the kinetic and transport parameters [73].

Well known parameter estimation theory and methods have been applied only recently to large-scale polymerization reactors models [112, 4]. In the particular case of LDPE tubular reactors models, the parameter estimation problem turns out to be so large and complex that it is usually simplified, using heuristics based on the knowledge of the kinetic mechanism

[20, 19, 72]. In this section, we make use of the computational strategies described in Chapter 4 to solve large estimation problems arising in the development of LDPE reactor models.

8.2.1 Model Structure

The first-principles reactor model under consideration describes the gas-phase free-radical copolymerization of ethylene with a comonomer (e.g. vinyl acetate) in the presence of several different initiators and chain-transfer agents (CTAs) at supercritical conditions. The mechanism postulated to describe the copolymerization kinetics is presented in Figure 8.2. Here, the symbols I_i with $i \in \{1, \dots, N_I\}$, R , M_1 , M_2 and S_i with $i \in \{1, \dots, N_S\}$ denote the initiators, radicals, monomer, comonomer and CTA molecules, respectively. The symbol η_i represents the efficiency of initiator i . The symbols $P_{r,s}$ represent "live" polymer chains ending with a monomer unit; with r monomer units and s comonomer units. Similarly, $Q_{r,s}$ are "live" polymer chains with r, s degrees of polymerization but ending with a comonomer unit and $M_{r,s}$ are "dead" polymer chains. The respective reaction rates for the monomers, initiators, chain-transfer agents and "live" and "dead" polymer chains can be obtained by combining the reaction rates of the elementary reactions describing their production and consumption.

We recognize that a complete description of the polymer chain molecular weight distributions requires an extremely large number of population balances for the polymer chains. To avoid this, the method of single moments is used to describe macromolecular properties of the copolymer [105, 73]. The method of moments is based on the statistical representation of the polymer average molecular weights and the compositional properties in terms of the leading moments of the number chain-length distributions of the "live" and "dead" polymer chains. In this model, the univariate number chain-length distributions for $P_{r,s}$, $Q_{r,s}$ and $M_{r,s}$ are considered. Accordingly, the moments of the number chain-length distributions

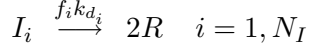
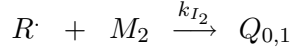
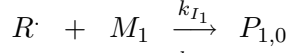
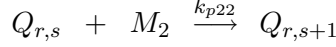
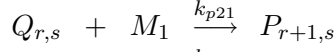
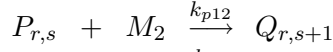
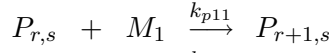
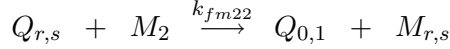
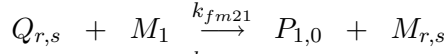
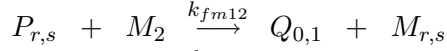
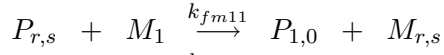
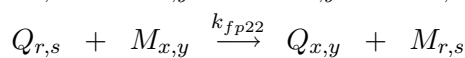
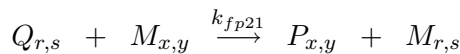
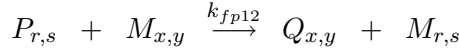
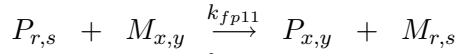
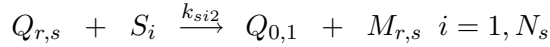
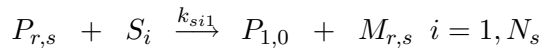
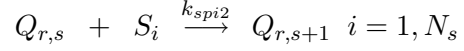
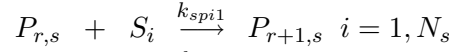
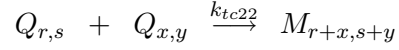
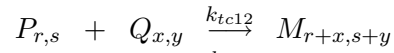
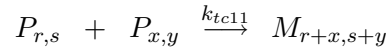
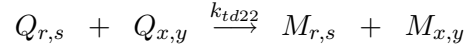
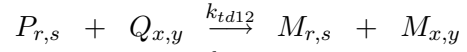
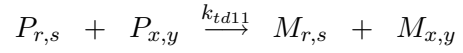
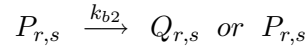
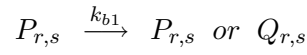
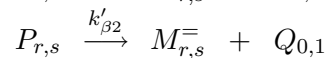
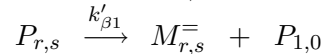
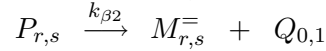
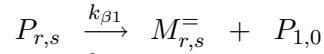
Initiator(s) decomposition

Chain initiation

Chain propagation

Chain transfer to monomer

Chain transfer to polymer

Chain transfer to CTAs

Incorporation of CTAs

Termination by combination

Termination by disproportionation

Backbiting

 β -scission of sec- and tert-radicals


Figure 8.2: Free-radical copolymerization mechanism of ethylene with a comonomer.

are defined as,

$$\lambda_n^i = \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} (r+s)^n R^i(r, s) \quad n \in \{0, 1, 2\}, \quad i \in \{1, 2\} \quad (8.1)$$

$$\mu_n = \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} (r+s)^n D(r, s) \quad n \in \{0, 1, 2\} \quad (8.2)$$

where $R^1(r, s) = [P_{r,s}]$, $R^2(r, s) = [Q_{r,s}]$ and $D(r, s) = [M_{r,s}]$. With this, the number- and weight-average molecular weights, the degrees of long chain branching (LCB) and short-chain branches (SCB) per 1000 carbon atoms can be expressed in terms of the leading moments of the univariate length-chain distribution.

The model complexity is often reduced by making some general validated assumptions such as: the reacting mixture forms a single supercritical phase, plug flow is observed along the reactor and net production rates of the radicals and "live" polymer chains are negligible [73]. Considering this, we can derive a set of molar and energy balances describing the evolution of the reacting mixture, the jacket water and of the wall temperature along the axial and time dimension. The design equations are presented in Appendix E. The equations for neighboring zones are coupled through material balances at the feed points. The model also presents complex *multi-point* boundary conditions due to the presence of concurrent jacket water flows and due to the coupling of heat fluxes along the zone joints (Figure 8.1). The complexity of the boundary conditions is a remarkable issue since it limits the application of DAE solvers for their solution. It is commonly assumed that the LDPE reactor is at a *quasi-steady-state* at all times. This is a reasonable assumption if the jacket and wall do not undergo any dynamic transitions [56]. In dynamic scenarios, the reactor core can still be assumed to be at steady-state since its time constant is small compared to that of the wall and the jacket [56].

The accuracy of the tubular reactor model depends significantly on the prediction of the reacting mixture properties. The model under study contains a large number of algebraic equations required for the calculation of the physical, thermodynamic and transport properties. The properties that have more influence on the model accuracy are the mixture density,

heat capacity and viscosity. The gas phase density and heat capacity are obtained through the Lee-Kesler equation of state [79]. The rest of the properties are obtained from complex semi-empirical correlations. All the correlations used have been validated experimentally and are reported in [73].

8.2.2 Model Implementation

To represent the overall dynamic model of Appendix E in general terms, we collapse all the partial differential and algebraic equations (PDAEs) corresponding to conservation equations, thermodynamics, transport, and kinetic expressions for all the reactor zones into a single set of PDAEs. This can be done by grouping the states corresponding to all zones into a single variable vector and defining the continuity equations between zones as algebraic equations. Following this reasoning, we represent the multi-zone reactor model as,

$$\frac{\partial z}{\partial t} + \nu(t, x) \frac{\partial z}{\partial x} + \kappa(t, x) \frac{\partial^2 z}{\partial x^2} = f_z(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.3a)$$

$$\frac{\partial w}{\partial x} = f_w(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.3b)$$

$$0 = f_y(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.3c)$$

$$\gamma(t, x) = \chi(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.3d)$$

$$z(0, x) = z^0(x) \quad (8.3e)$$

where $z(t, x) \in \mathbb{R}^{n_z}$ is used to represent the differential states in space and time with distributed initial conditions $z^0(x)$. These states correspond to the cooling water temperature and the reactor wall temperature at all zones. Symbol $w(t, x) \in \mathbb{R}^{n_w}$ denotes differential states in space such as the reacting mixture temperature, the molar flow rates of gaseous components, the chain moments, among others. Symbol $y(t, x) \in \mathbb{R}^{n_y}$ denotes the algebraic states corresponding to the rest of the model variables such as the cooling water and reacting mixture velocities, densities, heat capacities, among others. Symbol $p(t)$ denotes time-varying parameters used to account for unmodeled effects and uncertainty. Symbol $u(t)$ denotes the model inputs corresponding to side-stream inlet temperatures and flow

rates, among others. In the actual reactor, the inputs can only be fed at the beginning and end of each reactor zone. Consequently, no explicit dependence on the internal spatial dimension x is considered. For convenience, we will define a set of variables $\gamma(t, x) \in \mathbb{R}^{n_\gamma}$ that map all the model states and inputs into a set of output and input variables of relevance in the actual LDPE process. The boundary conditions of system (8.3) can be expressed in the general form,

$$0 = \varphi \left(z(t, 0), z(t, x^L), \frac{\partial z}{\partial x}(t, 0), \frac{\partial z}{\partial x}(t, x^L), w(t, 0), u(t) \right). \quad (8.4)$$

On average, the dynamic reactor model will contain 3 PDEs, 20 ODEs and 500 AEs. The equations are defined over very long axial horizons (1-2 km) containing all the zones and time horizons of less than one hour.

LDPE reactors present steep temperature profiles in space. In addition, the reactor presents steep profiles for some gaseous components such as the initiators that are totally consumed immediately after they are injected into the reaction zones. On the other hand, the quasi-steady-state assumption for the reactor core eliminates stiffness problems along the time dimension. Motivated by these observations, we use a fine discretization mesh in space and a coarse mesh in time. A finite element discretization scheme at Radau collocation points is performed along the space dimension. Since in Radau collocation the last collocation point lies at the boundary of each zone, this discretization scheme is particularly useful in order to handle the multi-point boundary conditions arising in the jackets. We use an implicit Euler discretization in time. In Figure 8.3, we present a conceptual representation of the discretization approach for the reactor and jacket temperature profiles of a particular zone.

After full discretization, we implemented the LDPE reactor model in algebraic form in the modeling platform AMPL. The model is implemented in superstructure form as sketched in Figure 8.1. In other words, the same model equations are used in both the cooling and the reaction zones. In a cooling zone, the ethylene/CTA side streams and initiator flows are set to zero. This model representation has been useful in implementing different reactor designs

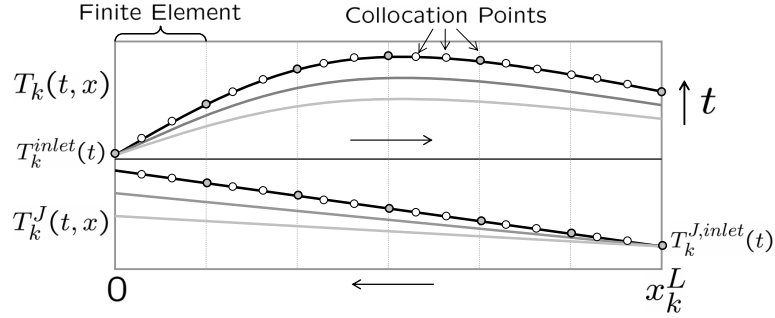


Figure 8.3: Schematic representation of discretization approach. Reactor core (top) and Jacket (bottom).

and configurations. In addition, the discretization scheme can be adapted as required in order to improve the accuracy and/or to obtain faster solutions.

8.2.3 Model Parameters

A common observation in most literature reports is the *lack of a consistent data base of kinetic parameters* for LDPE reactor models. In some cases, the reported kinetic parameter values vary by several orders of magnitude [73]. Because of this, it is often necessary to re-estimate the parameters using experimental data from the particular laboratory or industrial reactor under study [20, 19].

Industrial LDPE tubular reactors are also subject to persistent and uncertain variability that needs to be considered if industrial data is used for model tuning. This requires the selection and estimation of parameters to account for this. One of the most interesting problems associated with the operation of LDPE tubular reactors is the persistent fouling of the inner reactor wall. This phenomenon depends on many factors and it is difficult to predict by means of simple mechanistic models [23]. A simple engineering way to handle this problem is to infer on-line the heat transfer coefficients (\mathbf{U}^{in}) from available temperature measurements.

As illustrated in Figure 8.1, a number of sidestream feeds of initiator mixtures (cocktails) are distributed along the reactor. At each feed point, a typical mixture can include up to four different initiators with different chemical properties. These initiators decompose to generate the radicals that start the polymerization. The initiator decomposition reactions include sets of complex reaction subnetworks involving the formation of highly active intermediate species that can react among each other or with impurities in the reacting mixture before generating the desired radicals. Therefore, there is an efficiency factor η_i associated with the decomposition of each initiator. These initiator efficiencies depend on the reacting mixture temperature and pressure, the degree of mixing at the feed points, and the presence of other species such as impurities or CTAs. Moreover, the efficiency of an individual initiator might vary with its concentration in the reacting mixture [84, 111]. In LDPE reactors wide variations of the reacting mixture temperature, pressure, composition and physical properties are observed. As a consequence, wide variations of the efficiencies are expected as well along the reactor and over time due to the accumulation of impurities. To account for this, the initiator efficiency for each reaction can be also be estimated for each reaction zone in order to match the plant reactor temperature profile. Previous studies have shown satisfactory results using this approach [74, 72]. However, in order to simplify the parameter estimation task, it is usually assumed that there is a common efficiency for all the initiators in the mixture. While this assumption provides sufficiently accurate model predictions, it is expected that the estimation of the individual initiator efficiencies will result in a better match of the reactor temperature profile.

8.2.4 Case Studies

The available measurement sets contain snapshots of the LDPE reactor behavior at nearly stationary conditions. Consequently, we will assume that the LDPE reactor is at a *quasi-steady-state* at all times. With this, the resulting LDPE reactor model is reduced to 525 DAEs defined over a long horizon in the axial space dimension (2-3 km). The available

Measurement	Input	Output	On-Line Sensor	Laboratory
Inlet Pressure	x		x	
Jacket Inlet Temperatures	x		x	
Sidestreams Inlet Temperatures	x		x	
Ethylene Sidestream Flow Rates	x		x	
Comonomer Sidestream Flow Rates	x		x	
CTA Sidestream Flow Rates	x		x	
Initiator Flow Rates	x		x	
Temperature Profile		x	x	
Overall Conversion		x	x	
Jacket Outlet Temperatures		x	x	
Melt Index		x	x	x
Weight-Average MW (M _{ww})		x		x
Number-Average MW (M _{wn})		x		x
Polymer Density		x		x
Long-Chain Branching		x		x

Table 8.1: Summary of available measurements in industrial LDPE reactors.

measurement information is summarized in Table 8.1. As can be observed, there exists limited on-line measurement information to monitor the performance of LDPE reactors. In particular, measurements of the polymer macromolecular properties are seldom available. Because of this, the polymer properties are usually inferred on-line through the reactor temperature profile and melt index measurements. A typical industrial reactor can easily contain around a hundred thermocouples to monitor the temperature profile.

The main objective of the parameter estimation procedure is to find the best set of on-line adjustable parameters and kinetic parameters able to fit the reactor operating conditions under the available scenarios. We use the standard least-squares and EVM formulations of Chapter 4. In both formulations, the number of DAE constraints increases as we add data sets into the problem. For instance, a problem with 5 data sets can contain up to 2500 DAEs. The DAE model is discretized in space using Radau collocation on finite elements. A total of 300 collocation points was necessary to achieve a good degree of accuracy. The resulting discretized model contains around 11,000 highly nonlinear algebraic equations. This gives

rise to a large-scale and sparse NLP. The resulting NLPs problems were implemented in AMPL are solved with IPOPT. AMPL provides first and second derivative information to the solver. We perform an inference analysis study on the estimated parameters using reduced Hessian information extracted from IPOPT. In addition, we overcome memory bottleneck limitations for problems with many data sets using the previously proposed Schur decomposition strategy.

Estimation of On-line Adjustable Parameters

As a first case study, we consider the estimation of the initiator efficiencies and heat transfer coefficients for different data sets. The parameters are estimated in order to match the reactor temperature profile and the jackets inlet and outlet temperatures (e.g. close the energy balance). This implicitly predicts the reactor conversion and polymer properties.

We compare our estimation approach with a simplified estimation approach currently used as industrial standard [74, 72]. In the simplified approach, a single efficiency is assumed for the initiator mixture in every zone. Each of these efficiencies is assumed to be constant along the zone. The heat transfer coefficients (U_k^{in}) are estimated for every cooling and reaction zone and are assumed to be constant along each zone as well. Furthermore, the parameters are estimated sequentially along the reactor, that is, solving *one zone at a time*. In addition, *ad-hoc* simplifications are required in order to handle the multi-point boundary conditions arising in the jackets and thus allow the use of standard DAE solvers. This approach is followed in most studies because the resulting estimation problems are relatively small and simple to solve [74, 72]. In the proposed approach, we estimate all the individual efficiencies and HTCs for all the reactor zones *simultaneously*. With this all-at-once approach we can capture *downstream interactions* of the parameters and measurements along the reactor. In other words, we can incorporate more information into the problem. As a consequence, this approach is expected to yield more reliable parameters. However, this comes at the expense of having to solve much larger estimation problems.

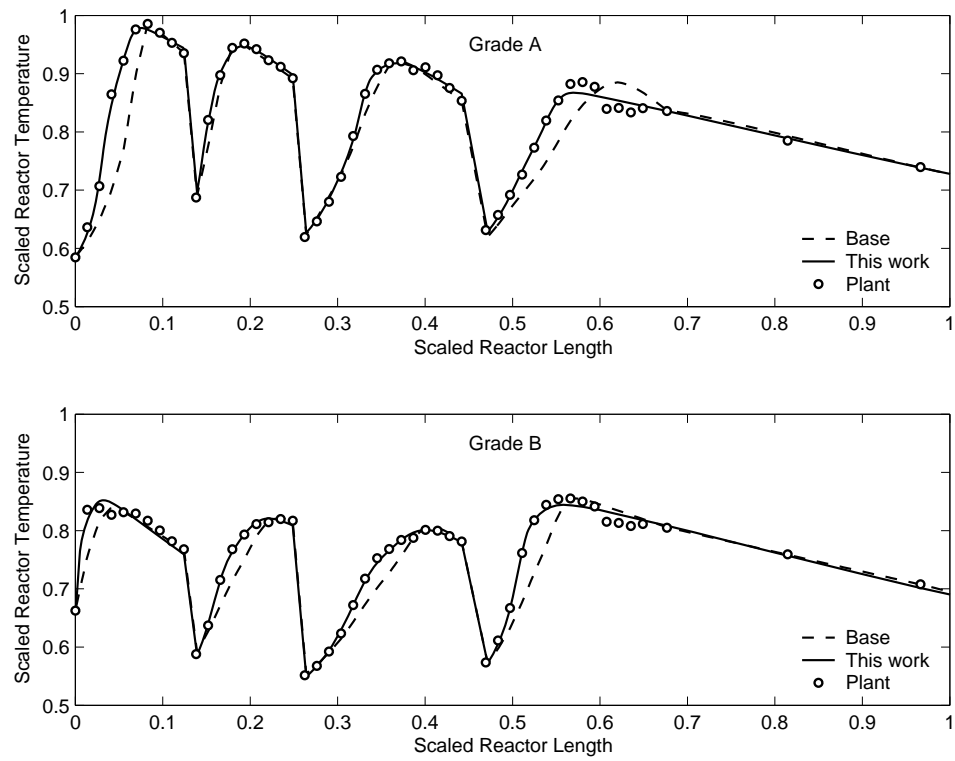


Figure 8.4: Plant and predicted temperature profiles using the base and the proposed estimation strategies for two different grades.

Grade	Base	This work	% Improvement
A	13262.28	6628.43	50.02
B	17474.44	4458.40	74.48

Table 8.2: Optimal objective function values for estimation approaches. Grades A and B cases.

We consider the production of two different polymer grades (A and B) in the same reactor. Grade A is a copolymer with high comonomer content and grade B is a homopolymer, both of them of high molecular weight. The reactor under study contains 13 zones but, for confidentiality reasons, we only present the first few reaction zones. In Figure 8.4 we illustrate the plant and predicted profiles using the base and the proposed all-at-once approaches. It is evident that the consideration of the individual efficiencies in the all-at-once approach has a strong impact on the initial shaping of the profiles in the reaction zones. This is due to the fact that the individual contribution of each initiator can now be considered at different temperature levels [84]. In addition, the all-at-once approach can handle the multi-point boundary conditions more efficiently and thus it allows to close the reactor energy balances more accurately. In Table 8.2 we contrast the optimal objective function values for both estimation approaches. The objective function is the sum of squares of the deviations between the plant and predicted temperatures along the reactor and jackets. As can be seen, the proposed approach yields a closer match of the data. In Figure 8.5 we present the conversion predicted by the model using both estimation approaches. Twenty different grades, considering wide ranges of operating conditions and polymer properties are presented in the graph. The proposed approach gives more consistent and accurate predictions. For the twenty grades considered, the average conversion deviation was reduced from 12.1% using the base case approach to 2.5% with the proposed all-at-once approach.

In order initialize the NLPs of the all-at-once approach, we used the solutions obtained from the sequential zone-by-zone strategy. The zone-by-zone estimation of the on-line pa-

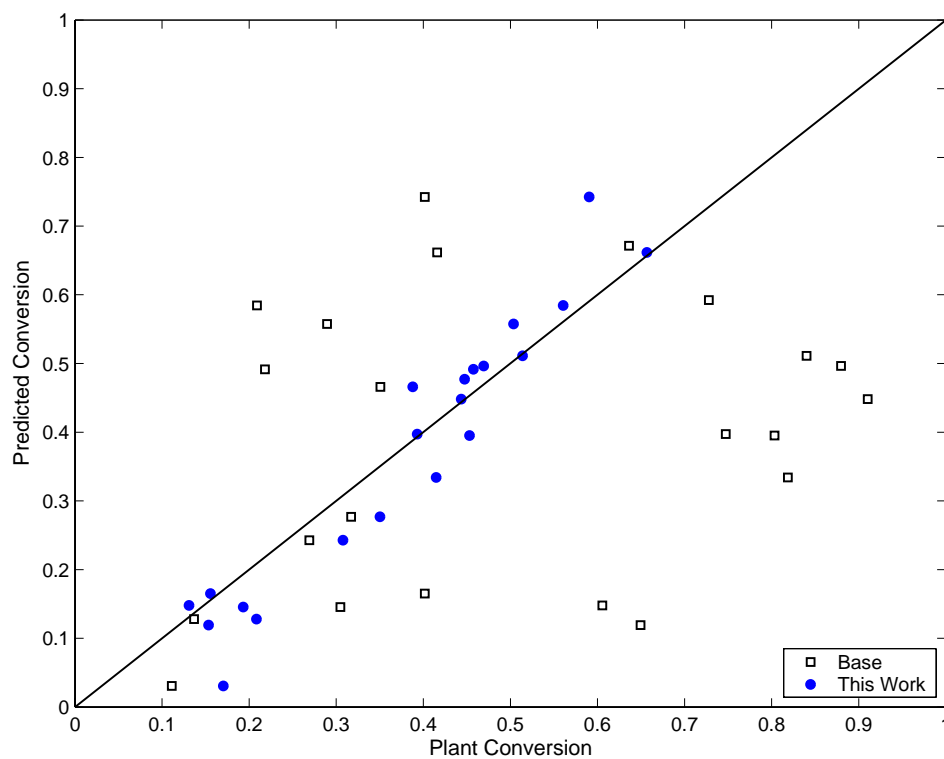


Figure 8.5: Plant and predicted conversions for estimation approaches analyzed. Results for twenty different grades.

rameters the sequential zone-by-zone strategy required 20-30 CPUs. Around 80-90% of the solution time was spent for the integration of the DAE model for every zone. However, this approach was found to be expensive for the solution of the overall estimation problem since, in this case, the entire reactor DAE model needs to be integrated at each iteration. The full integration takes around 4.5 CPUs per iteration. Following this reasoning, the sequential approach was expected to become expensive in the solution of more complicated estimation problems and it is was not considered further in this study. Significant reductions on the overall CPU time were obtained using the proposed all-at-once approach. Here, we use linear solver MA27 to factorize the KKT matrix at each iteration in IPOPT. The problems are solved on a 3.0 GHz, 1 Gb RAM, Pentium IV PC. Since a good starting point is provided and the inequality constraints are never active at the solutions, a small initial barrier parameter μ of 1×10^{-6} was set for all calculations. In all cases, the reduced Hessian matrix was found to be positive definite at the solution with large eigenvalues ranging from 10^2 to 10^{10} . Accordingly, inertia correction was not necessary at the solution and we can conclude that the parameters are unique. The computational results for Grades A and B are presented in Table 8.3. The NLPs are quite large, containing around 11,000 constraints and 32 degrees of freedom (corresponding to the entire set of on-line adjustable parameters). Nevertheless, the solution approach is fast, taking around 1.5 CPUs per iteration. Note also from Table 8.3 that the KKT matrix is very sparse. For instance, if we take the number of nonzeros in the Jacobian of the first case and divide this by the number of constraints, we see that the Jacobian has around 14 nonzero entries per row (a total density of $\approx 0.13\%$).

Estimation of Kinetic Parameters

The all-at-once estimation approach provides accurate matches of the reactor temperature profile. With this, we can predict the reactor conversion and the macromolecular properties. To refine the predictions of these properties, we estimate the kinetic constants (Π in (4.1) and (4.2)) over different ranges of operating conditions. The kinetic rate constants presented

Grade	Constraints	Parameters	LB	UB	Iterations	CPU's	NZJ	NZH
A	11955	32	374	361	11	17.03	166425	87954
B	11283	32	374	361	8	10.06	138666	76890

Table 8.3: Computational results for single-set NLP problems. On-line parameter estimation case studies. LB=number of lower bounds, UB=number of upper bounds, NZJ=number of nonzeros in Jacobian, NZH=number of nonzeros in Hessian.

in Figure 8.2 are functions of the reacting mixture temperature and pressure. These have the following form,

$$k_i = k_i^0 \exp \left[-\frac{\Delta E_{a_i} + P \Delta E_{v_i}}{RT} \right] \quad (8.5)$$

where subindex i belongs to the entire set of elementary reactions in the kinetic mechanism. Symbol k_i^0 denotes the pre-exponential factor, ΔE_{a_i} the activation energy, ΔE_{v_i} the activation volume and P the reactor pressure. A large portion of the model nonlinearity comes from the multiple Arrhenius expressions. Therefore, the estimation of kinetic parameters is a much more computationally intensive problem.

Since the number of kinetic parameters is large, the estimation procedure has been traditionally decomposed into subproblems with fixed subsets of parameters [72][20][19]. For instance, a first step is to estimate the propagation and termination kinetic rate constants to match the reactor overall conversion. The second step consists of estimating other kinetic parameters to match macromolecular properties such as number- and weight-average molecular weights, LCB, and polymer density. This iterative approach is repeated until the best set of parameters is obtained. In the all-at-once approach we include multiple data sets over wide ranges of operating conditions in order to estimate the whole set of kinetic parameters and on-line adjustable parameters. Consequently, this approach is expected to give more reliable parameters but it also leads to highly complex problems.

We consider the estimation of the kinetic parameters for homopolymerization reactions.

The homopolymer grades are produced on the same reactor operating over different ranges of temperature, pressure, and concentration of CTA. For all the problems solved, the entire set of on-line and kinetic parameters are estimated to match the reactor and jacket temperatures, overall reactor conversion, number- and weight-average molecular weights, polymer density and degree of long-chain branching (LCB). Nominal values of the macromolecular properties are used. The motivation behind systematic estimation strategies becomes clear from Figure 8.6. Here, two different 95% ellipsoidal confidence regions are presented for the pre-exponential factor and activation energy of the propagation rate constant (k_{p11}). The ellipsoids were obtained from the solution of standard least-squares estimation problems with one and three data sets. It is clear that there is a large uncertainty associated with the point estimation of the pre-exponential factor. This uncertainty can be reduced significantly by adding more data sets in the problem. On the other hand, note that a single data set is informative enough to obtain a tight confidence interval for the activation energy. This can be explained from the fact that the temperature profile along the reactor involves a *wide* range of temperatures. In other words, the measurements seem to be *highly informative*. In Figure 8.7 we present the confidence regions for problems with three and six data sets. The ellipsoid obtained from the six data set problem is tighter, but the relative reduction is not as dramatic as in the previous case.

Multi-set estimation problems were solved using the standard least-squares and EVM formulations. For standard least-squares, we estimate 32 on-line adjustable parameters for each data set along with 28 kinetic parameters. The heat of polymerization is estimated as well. For the EVM formulation, we have many more degrees of freedom. In particular, the degrees of freedom for every set are increased by 52 input variables corresponding to the multiple measurements distributed along the reactor. In all cases, the estimation problems are initialized using the values of the on-line adjustable parameters obtained in the previous section. The computational results are presented in Table 8.4. Note that the number of iterations remains consistent among the different problems. The availability of exact first

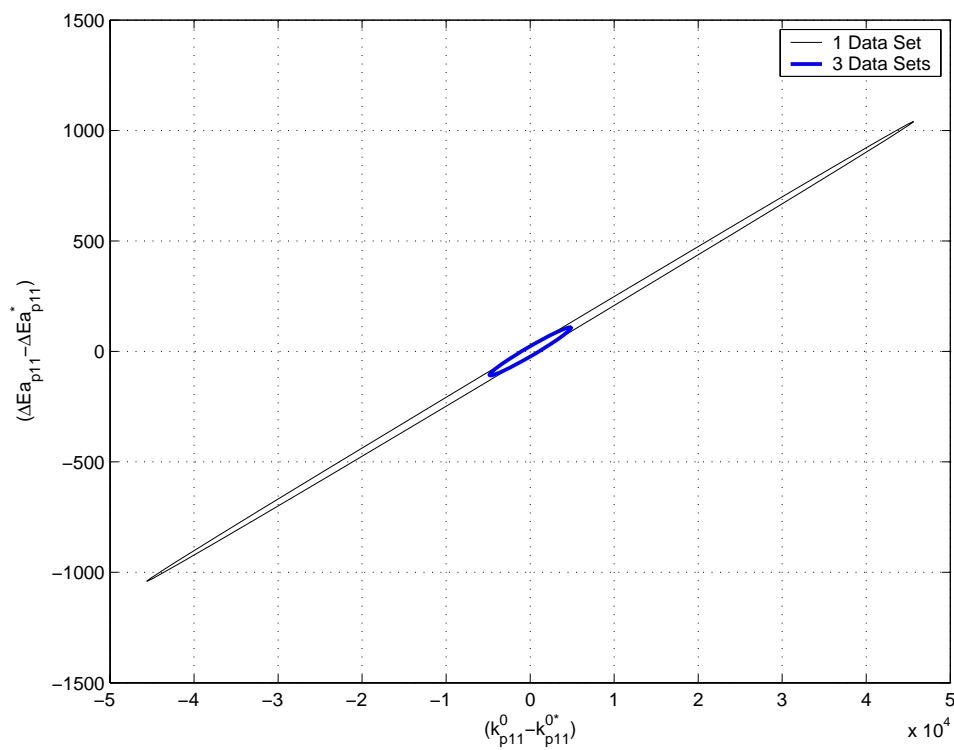


Figure 8.6: Confidence regions for the parameters of the propagation rate constant k_{p11} . Results with 1 and 3 data sets.

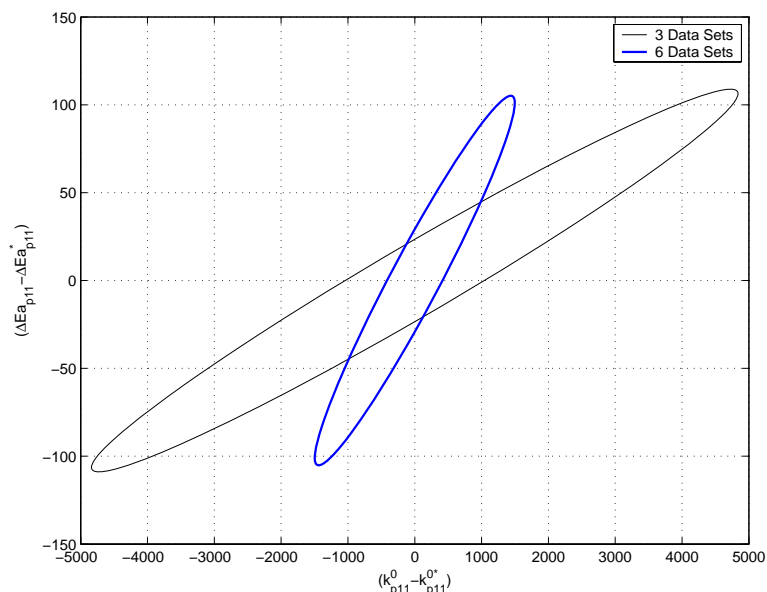


Figure 8.7: Confidence regions for the parameters of the propagation rate constant k_{p11} . Results with 3 and 6 data sets.

and second order derivative information is crucial to achieve this efficiency. Note also that the sparsity of the KKT matrix (number of nonzeros in Jacobian and Hessian) remains unaltered between the least-squares and the EVM formulations. This explains the similar computational times required by the algorithm despite the dramatic increase of the number of degrees of freedom in the EVM formulation. The largest estimation problem was solved in less than 20 minutes on a 3.0 GHz, 1 Gb RAM, Pentium IV PC. Problems with more data sets were not solved in this preliminary study due to limitations in computer memory requirements arising in the factorization of the KKT matrix. In particular, it was found that the minimum degree reordering strategy used by MA27 introduces a large degree of fill-in in the factors of the KKT matrix.

The model predictions were notably improved using the new set of parameters. In Figure 8.8 we compare the plant and model predictions for the overall reactor conversion, number- and weight-average molecular weights and degree of LCB. We considered a total of six

Data Sets	Constraints	DOF	LB	UB	Iterations	CPUs	NZJ	NZH
3	33900	121	1246	1207	68	451.51	520275	552738
3 (EVM)	33952	277	1366	1327	57	345.82	520636	553080
6	68421	217	2467	2389	58	900.21	1058412	1119258
6 (EVM)	68627	529	2653	2575	71	1010.74	1059512	1119780

Table 8.4: Computational results for multi-set NLP problems. Homopolymerization case study. LB=number of lower bounds, UB=number of upper bounds, NZJ=number of nonzeros in Jacobian, NZH=number of nonzeros in Hessian, DOF=degrees of freedom.

	Conversion(%)	Mwn(%)	Mww(%)	LCB(%)	Density(%)
Base Parameter Set	1.49	23.24	18.58	19.20	0.0965
New Parameter Set	0.12	6.20	3.31	6.27	0.0875

Table 8.5: Average deviations between plant and model predictions for reactor conversion and grade macromolecular properties for 14 different grades. Homopolymerization case study.

estimation and eight validation cases including different reactor configurations and wide ranges of operating conditions. These grades span a range of 30-37% on the overall reactor conversion, 14500-19500 g/gmol on the number-average molecular weight, 150,000-450,000 g/gmol on the weight-average molecular weight, and 2.4-2.8 on the number of LCBs per 1000 carbon atoms. Average values of the deviations between the model predictions and the plant measurements for different polymer properties are presented in Table 8.5. We compare the model deviations using the base set of kinetic parameters provided from industry and the new parameters. As can be seen, the new predictions outperform those obtained using the base set in all cases.

In all the estimation studies, we assumed equal initiation and propagation rate constants

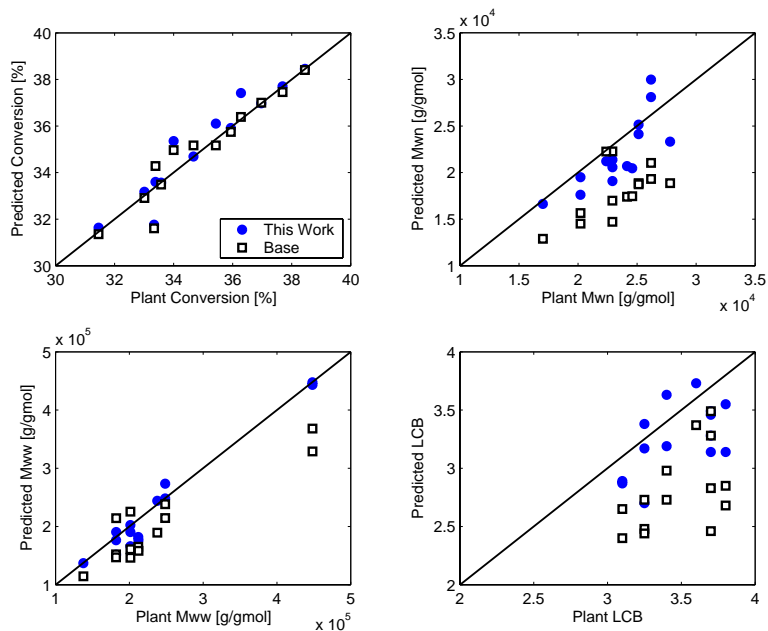


Figure 8.8: Homopolymer grade macromolecular properties. Plant and model predictions.

[73]. The rate constants for termination by combination and disproportionation were also assumed to be the same. We have found that the simultaneous estimation of the initiator efficiencies and the initiator decomposition rate constants leads to non-unique solutions (e.g. they cannot be estimated independently). Therefore, the decomposition rate parameters were fixed to their nominal values provided by peroxide vendors. Most of the estimated parameter values lie well between reported ranges. However, it has been found that the estimated activation volumes for the propagation, chain transfer to monomer and back-biting reactions have *opposite signs* to those reported in the literature [73]. With the base set of parameters the model could not predict the effect of large pressure variations on the weight-average molecular weight. With the new set of estimated parameters, the model was able to predict this pressure effect but some of the estimated activation volumes have opposite signs. We believe that this is due to a structural limitation of the model. For instance, decreased polymer solubilities in the gas phase due to large pressure fluctuations

are not taken into account in current model structure. It seems that the activation volumes are compensating for this. It has also been found that the estimated β -scission reaction rates present wide confidence intervals. These parameters can only be estimated reliably if measurement information of the polymer vinylenes and vinylidenes content is included in the problem. Unfortunately, this information was not available for this study.

Exploitation of Multi-Set Structure

As we have seen, it is possible to obtain more reliable parameters as we incorporate measurement information into the problem. On the other hand, we have also seen that the associated NLPs become quickly intractable due to the size and complexity of the LDPE reactor model. The key bottleneck is the factorization the KKT matrix. Here, we apply the parallel decomposition strategy of Chapter 4 to overcome these limitations [132].

To formulate the EVM estimation problem (4.2) as a multi-scenario NLP of the form in (4.4), the variables are separated by data sets or scenarios k . For the implementation of this strategy, each data set is defined as a separate NLP model in AMPL which provides the required derivative information for each instance. Inside each model, we *indicate* the set of variables corresponding to the global parameters Π . This is required to build the linking variables vector inside IPOPT.

In Figure (8.9), we present computational results associated with the solution of multi-scenario NLPs. The results were obtained in a Beowulf-type cluster using standard Intel Pentium IV Xeon 2.4Ghz, 2Gb RAM processors running on Linux. The parallel results are compared against those obtained from the direct factorization of the multi-scenario problems in a single processor. The MA27 linear solver with a minimum degree ordering strategy was used for all the sparse factorizations. The dense linear solver DGESV from the LAPACK library was used to factorize the Schur complement.

As can be seen, the solution of the multi-scenario NLPs through the direct factorization

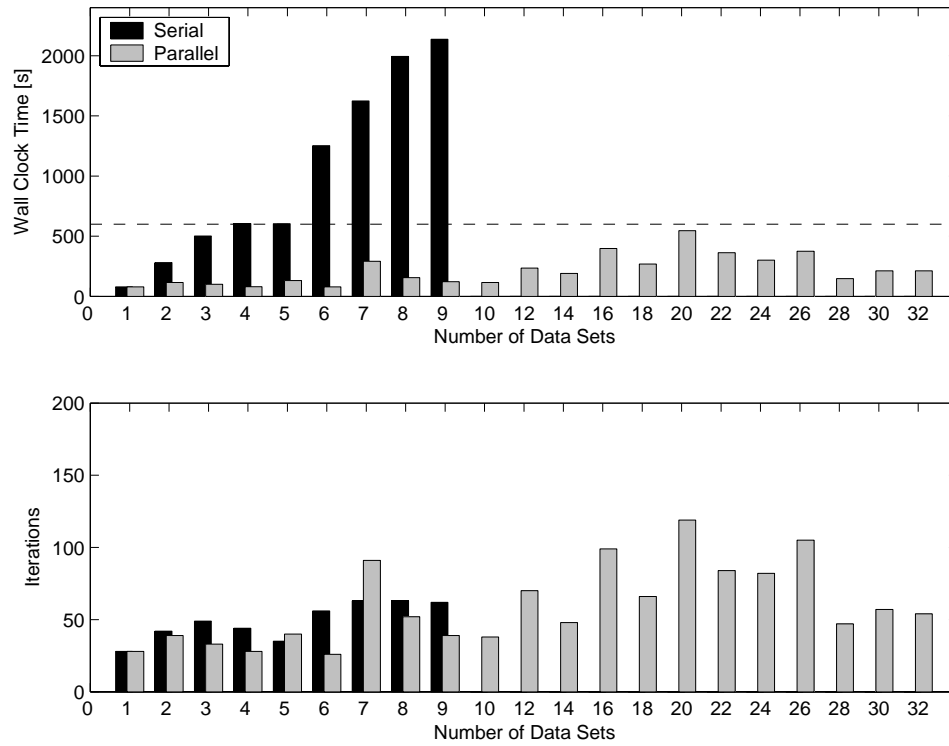


Figure 8.9: Total time and number of iterations for the solution of multi-scenario NLPs with IPOPT. Direct factorization and parallel implementations.

approach becomes intractable after 9 data sets. In contrast, the parallel implementation is able to overcome this memory bottleneck allowing the solution of problems with over 32 data sets. Here, the largest NLP contains around 399,632 constraints and 2,170 degrees of freedom. Note that the solution time increases significantly in the direct factorization approach as we add more data sets. In particular, it is interesting to observe that the complexity of this strategy scales almost *quadratically* with the NLP size. The solution of a 9 data set problem with a direct factorization takes more than 30 minutes. In contrast, the parallel decomposition takes consistently less than 10 minutes regardless of the number of data sets.

Note that the solution times and number of iterations do not seem to follow any particular trend, presenting "random" jumps as we add or remove data sets. It is important to emphasize that this behavior is due to the complexity of the problem and not to the convergence properties of the optimization algorithm (e.g. the solution of the 32 data set problem requires fewer iterations than that with 20 data sets). This behavior is mainly attributed to the nonlinearity of the constraints and ill-conditioning of the KKT matrix. In particular, the high nonlinearity of the constraints gives rise to directions of negative curvature. Because of this, different number of inertia correction steps (e.g. more factorizations per iteration) are required for different problems. In order to avoid this, we present in Figure (8.10) computational results on the time required per iteration and per factorization of the KKT matrix as a more consistent measure of the scalability of the decomposition strategy. Note that the effects of the problem complexity are still reflected in the time required per iteration but not on the time per factorization. Nevertheless, the time per iteration can be consistently kept below 5 seconds while the factorization in the direct factorization approach can take as much as 35 seconds before running out of memory.

In Figure 8.11 we present a summary of the computational results for both the direct factorization and the parallel decomposition approaches. From the direct factorization results, it is possible to observe that the number of iterations taken by the interior-point algorithm

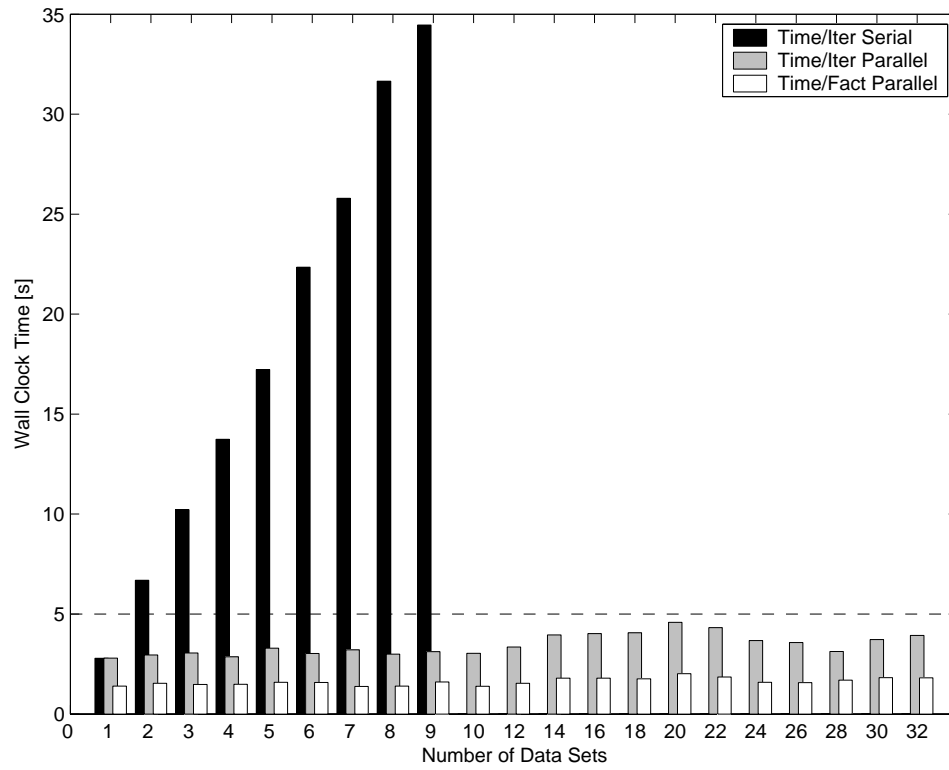


Figure 8.10: Time per iteration and per factorization of the KKT matrix during the solution of multi-scenario NLPs with IPOPT. Direct factorization and parallel implementations.

is not affected as we add degrees of freedom to the estimation problem. It is important to emphasize that this desired convergence properties can be obtained with full-space solvers using *exact* derivative information. As the proposed parallel decomposition strategy does not alter the core NLP algorithm in IPOPT, these desired convergence properties are retained.

8.3 Nonlinear Model Predictive Control

In this section, we present a Nonlinear Model Predictive Control (NMPC) application to optimize the operation of LDPE reactors. We start with a traditional design in which the controller is used to reject process disturbances and keep the reactor outputs at targets. We then describe an economics-oriented NMPC/D-RTO design in which the controller optimizes the overall production rate while rejecting the process disturbances.

The typical decision-making hierarchy in industrial LDPE processes consists of a target setting layer in which an operator receives the production schedule of different polymer grades. The operator sets the temperature profile of the reactor that is known by experience to give the desired polymer properties (e.g. melt index). The temperature set-points are communicated to multiple PID controllers distributed along the reactor that try to keep the temperature profile at the desired target. The main tasks of the regulatory control layer is to reject short-term disturbances and to follow the temperature profile set-points provided by the operator during grade changes.

As shown in Figure 8.12, the PID controllers are normally grouped by zones in order to regulate the *local* temperature profile. Input variables such as the initiator flow, the jacket inlet temperature, the jacket inlet flow, and the side stream temperatures can be manipulated independently by each local set of controllers. The fouling onset is, in particular, a difficult disturbance to reject. As the reactor fouls, the controllers need to keep the temperature profile at target. Because of this, they will tend to compensate by drop-

	NLP Statistics				Serial			Parallel			
NS	m	DOF	LB	UB	It	θ_{total} (s)	θ_{it} (s)	It	θ_{total} (s)	θ_{it} (s)	θ_{kkt} (s)
1	12319	92	425	412	28	78.24	2.79	28	78.24	2.79	1.40
2	24638	159	850	824	42	280.86	6.69	39	115.22	2.95	1.54
3	36957	226	1275	1236	49	500.75	10.22	33	100.62	3.05	1.48
4	49276	293	1700	1648	44	604.45	13.74	28	80.08	2.86	1.48
5	61643	361	2126	2061	35	603.20	17.23	40	131.76	3.29	1.59
6	73962	428	2551	2473	56	1251.87	22.35	26	78.77	3.03	1.58
7	86953	495	2976	2885	63	1624.59	25.79	91	291.91	3.21	1.38
8	99944	562	3401	3297	63	1994.17	31.65	52	155.90	3.00	1.39
9	112935	629	3826	3709	62	2136.82	34.46	39	121.55	3.12	1.60
10	125254	696	4251	4121				38	115.34	3.04	1.39
12	137573	763	4676	4533				70	234.50	3.35	1.53
14	150564	830	5101	4945				48	189.94	3.96	1.79
16	200512	1098	6801	6593				99	398.35	4.02	1.79
18	225822	1232	7651	7417				66	268.03	4.06	1.76
20	251132	1366	8501	8241				119	545.19	4.58	2.02
22	275098	1500	9351	9065				84	363.14	4.32	1.85
24	300408	1634	10201	9889				82	301.13	3.67	1.58
26	325046	1768	11051	10713				105	375.00	3.57	1.57
28	349684	1902	11901	11537				47	147.14	3.13	1.69
30	374994	2036	12751	12361				57	212.31	3.72	1.81
32	399632	2170	13601	13185				54	212.15	3.93	1.81

Figure 8.11: Summary of computational results associated with the solution of multi-scenario NLPs with IPOPT. m =number of constraints, **DOF**=number of degrees of freedom, **LB**=number of lower bounds, **UB**=number of upper bounds, **It**=number of iterations, θ_{total} =total wall clock time, θ_{it} =wall clock time per iteration, θ_{kkt} =wall clock time per factorization of KKT matrix.

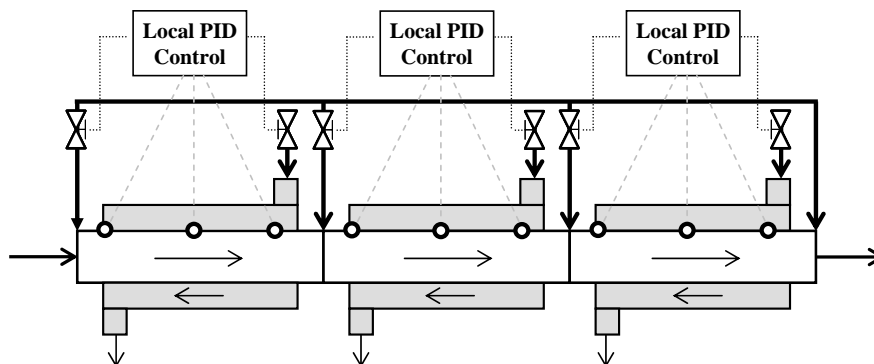


Figure 8.12: Regulatory control structure of LDPE tubular reactors.

ping the initiator flows and, implicitly, the production levels. It is important to emphasize that the controllers do not have any knowledge on the production levels of the reactor. Their objective is to keep the temperature profile at target which is set by the operator. Another problem that arises with a regulatory control architecture is that the controllers cannot foresee downstream interactions arising along the reactor. Because of this, the control of polymer properties at the reactor exit can become complicated. It is believed that a *centralized* model-based control strategy able to take all these interactions into account would achieve a much better performance. Such a strategy is illustrated in Figure 8.13. In particular, we are interest in exploiting the availability of detailed first-principles models to develop NMPC control strategies for LDPE reactors.

8.3.1 NMPC Problem Formulation

In the previous off-line parameter estimation study we assumed that the reactor was always at a quasi-steady-state. As we have seen, this is a reasonable assumption if the reactor jackets and wall do not undergo dynamic transitions. On the other hand, for NMPC we need to include the dynamics of the wall and of the jacket.

Consider the scenario in which the LDPE process is located at sampling time t_k . Here, the

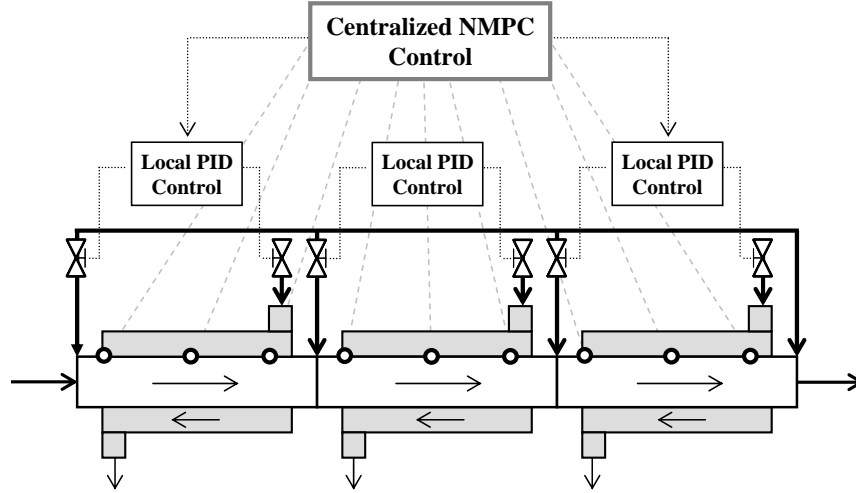


Figure 8.13: Centralized NMPC control structure of LDPE tubular reactors.

state of the process is given by at $z^k(x)$. Using this state, we would like to use the dynamic first-principles model to compute optimal policies for the controls $u(t)$ over a future horizon $[t_k, t_{k+N}]$ with sampling times of equal length $\delta = t_{k+1} - t_k$ that minimize a given performance index Φ_k . The NMPC problem formulation is given by,

$$\min_{u(t)} \quad \Phi^k \quad (8.6a)$$

$$\text{s.t.} \quad \frac{\partial z}{\partial t} + \nu(t, x) \frac{\partial z}{\partial x} + \kappa(t, x) \frac{\partial^2 z}{\partial x^2} = f_z(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.6b)$$

$$\frac{\partial w}{\partial x} = f_w(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.6c)$$

$$0 = f_y(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.6d)$$

$$\gamma(t, x) = \chi(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.6e)$$

$$0 \geq g(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.6f)$$

$$z(0, x) = z^k(x) \quad (8.6g)$$

$$0 = \varphi \left(z(t, 0), z(t, x^L), \frac{\partial z}{\partial x}(t, 0), \frac{\partial z}{\partial x}(t, x^L), w(t, 0), u(t) \right) \quad (8.6h)$$

$$t \in [0, N\delta].$$

From the solution of the NMPC problem we extract the current control actions $u^k = u_*(\delta)$.

The plant then evolves to the next state $z^{k+1}(x)$ and this is used to compute the next control action. In this analysis, we assume that the true state of the plant is equal to the prediction $z_*(\delta, x)$ so no state estimator will be necessary. In Section 8.4.1 we will discuss the specifics of a MHE estimator for LDPE processes. This estimator will be later coupled to the NMPC controller to provide output feedback.

We use a full discretization approach to solve the PDAE-constrained optimization problem (8.6). The inputs variables $u(t)$ are held constant at each time step. For the discretization of the reactor model, we use an average of 10 finite elements for the reaction zones and 2 finite elements for the cooling zones. Three collocation points are used. Upon spatial discretization, the PDAE reactor model translates into a DAE model containing around 9,000-10,000 DAEs in time. If 5 time discretization points are used, the DAE model is converted into an algebraic model with around 40,000-50,000 equations. We implement the NLP in AMPL and solve it with IPOPT. The MA57 linear solver is used to factorize the KKT matrix inside IPOPT. We use exact derivative information from AMPL. Moreover, we use a shifting strategy to warm-start subsequent NMPC problems and initialize the NLP solver with a small value of the barrier parameter $\mu = 1 \times 10^{-6}$. Nevertheless, in this section we assume that the NMPC problems can be solved instantaneously. In section 8.5 we present advanced step strategies to avoid computational delays.

8.3.2 Case Studies - Tracking NMPC

We first assume that the performance index is a tracking objective. With this, the NMPC controller seeks to minimize the transition time between the current state to a desired target. Therefore, we define the least-squares function,

$$\Phi^k = \int_0^{N\delta} \int_0^{x_L} [(\gamma(t, x) - \bar{\gamma}(t, x))^T \mathbf{Q}^{-1} (\gamma(t, x) - \bar{\gamma}(t, x))] dx dt. \quad (8.7a)$$

where \mathbf{Q}^{-1} is a diagonal weighting matrix used to penalize deviations from the references or targets. We will make use of the model variables $\gamma(t, x)$ to represent the inputs and

the controlled output variables with corresponding targets $\bar{y}(t, x)$. The targets are mapped to a set of output variables that can be measured in the process. In a typical LDPE process the outputs are usually the temperature profile, the polymer melt index and the polymer density. Here, we assume that the outlet temperatures of the jackets are not controlled variables but we impose physical bounds on them (e.g. reasonable temperature levels for cooling water going back to cooling tower). It is important to emphasize that the temperature profile is only controlled at discrete positions along the reactor (i.e. where the thermocouples are present). Nevertheless, having a first-principles model allows to control or impose constraints on temperatures at *unmeasured* positions. In addition, we can also control and/or impose constraints on unmeasured properties such as the polydispersity or molecular weights. The controls or manipulated variables include all the input variables of Table 8.1.

To test the performance of the tracking NMPC controller we consider a typical scenario arising in the operation of LDPE reactors. LDPE tubular reactors undergo periods where the polymer layer is *defouled* by means of pressure or thermal shocks [22]. Stabilizing the reactor under these disturbance cycles is crucial in order to avoid reactor runaway and to keep the polymer properties at target. To simulate the NMPC cleaning-fouling cycle, we ramp the heat transfer coefficients for all the reactor zones from their nominal value to +50% and then back to -20%. Since the controller cannot predict the behavior of the heat transfer coefficients, they act as exogenous disturbances. The plant response is obtained by simulating the reactor model using the current control action and the *true* heat transfer coefficients. Each time step corresponds to 5 minutes of operation. We use a prediction horizon of 5 time steps. The closed-loop simulations are run for a hundred time steps. The objective of the controller in this scenario is to keep the temperature profile as close as possible to a given reference profile as the reactor fouls and defouls.

In the top graph of Figure 8.14 we present the profiles of the heat transfer profiles for zones 2 and 3. Note that in the first stage the reactor gets clean so the values of the heat

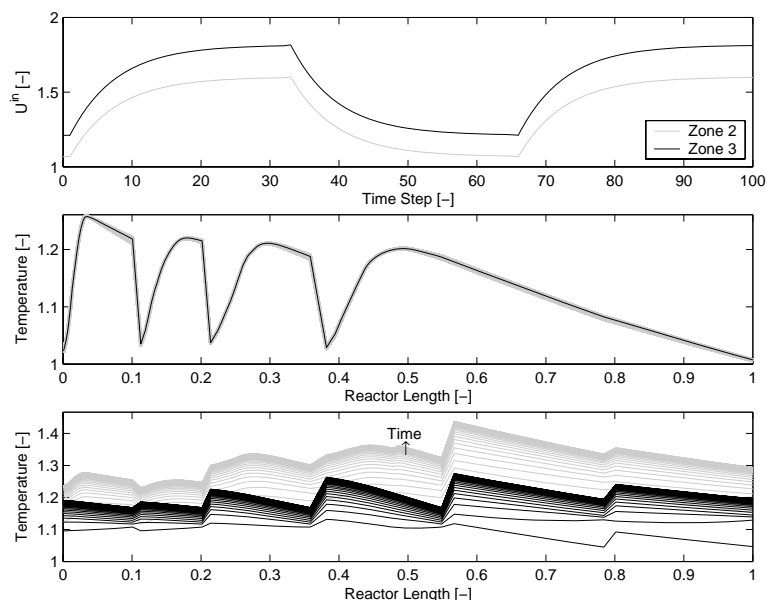


Figure 8.14: NMPC controller behavior. Ramping of heat transfer coefficients (top). Controller response for temperature profile (middle) - gray lines are the responses and dark line is the reference temperature profile. Responses of wall (gray) and jacket (black) temperatures (bottom).

transfer coefficient go up. The reactor then gets fouled and the heat transfer coefficient values go down. Note also that the reactor zones have different cooling capabilities. In the middle graph we present the closed-loop response of the temperature profile. The dark line is the reference profile and the gray lines represent the controller responses. As we can see, the controller is able to stabilize the system and keep the temperature profile close to the reference. The impact of the fouling-defouling disturbances can be appreciated in the bottom graph. Here, we present the dynamic responses for the wall and jacket temperatures for the first 35 time steps (defouling). The temperature levels raise significantly due to the decrease of the fouling layer thickness (increasing heat transfer coefficient). In other words, more heat can be dissipated to the jacket and walls. In addition, note that as the reactor defouls the wall temperature profile becomes similar in shape to that of the reactor core.

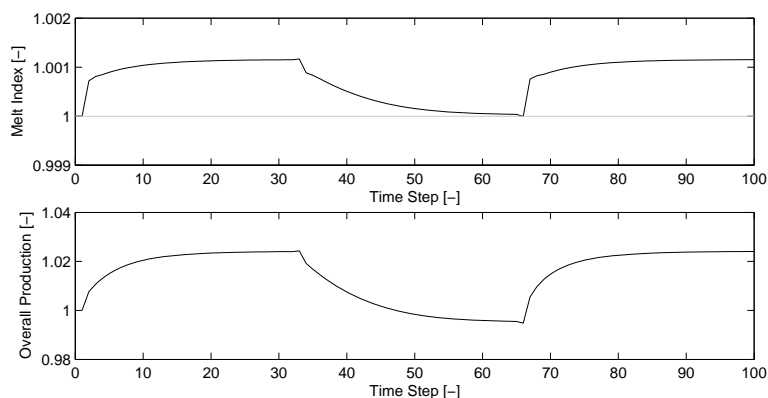


Figure 8.15: NMPC controller behavior. Closed-loop response of outlet melt-index (top). Response of reactor overall production (bottom).

In Figure 8.15 we present the reactor responses for the polymer melt index and the reactor overall production. In the top graph, we can observe that the controller achieves a tight control of the melt index. The melt index variability is less than 0.1% with respect to the reference value. Although a proper control of the reactor temperature profile is important to keep the melt index at target, the controller also makes use of the chain-transfer flow rate to help with the control. In the bottom graph we present the response for the reactor overall production. Note that the production increases as the reactor is cleaned. This results from the increase of the closed-loop initiator flows that keep the temperature profile at target. On the other hand, when the reactor fouls, the controller needs to drop the initiator flows and this in turn results in decreased production levels. It is important to emphasize that the controller objective is to keep the reactor at the desired temperature profile and not to control the production levels. In other words, the fluctuating production levels are an implicit consequence of the control actions.

In Figure 8.16 we present the closed-loop responses for some of the controller inputs. In the top graph we can see the response of the jacket inlet temperatures for zones 2 and 3. Note

that the controller responses do not follow obvious trends. This implies that there exists a relatively high degree of nonlinearity and coupling in the system. In particular, the jacket inlet temperatures have a strong impact on the outlet temperatures of the reaction zones (where the initiators are totally consumed) and on the temperature profiles of the cooling zones. In the middle graph we can see that the initiator flows follow a monotonic trend as we ramp the heat transfer coefficients. The controller decreases the amount of initiator during the fouling periods and increases it during the defouling periods. This is done in order to compensate for the fluctuating cooling capacity. Since the reactor temperature levels are largely determined by the heat of reaction, the initiator flows are used to control the local conversion levels in each zone. In the bottom graph we appreciate the closed-loop response of the the side-stream temperatures. The side-stream temperatures tend to have the strongest impact on the inlet temperatures at each zone. In addition to the initiator flows, the inlet side-stream temperatures play an important role in shaping the temperature profile at each zone. However, in this particular scenario, the controller does not make use of the side-stream temperature of Zone 3. This illustrates that the controller is implicitly taking into account interactions between zones.

8.3.3 Case Studies - Economics-Oriented NMPC

One would expect that the incorporation of a centralized NMPC controller would result in increased robustness and performance since it can better handle the multivariable interactions along the LDPE reactor. However, it is often difficult to appreciate the economic benefits of incorporating such a sophisticated controller. In principle, we could appreciate these benefits more easily if we would use a real-time optimization (RTO) layer in order to compute steady-state targets that optimize the process profitability and then use NMPC controller to obtain fast transitions between targets. However, this approach is difficult to apply to LDPE processes since they are seldom at steady-state due to frequent and persistent disturbances (e.g. fouling). In this work, we propose the incorporation of an economic

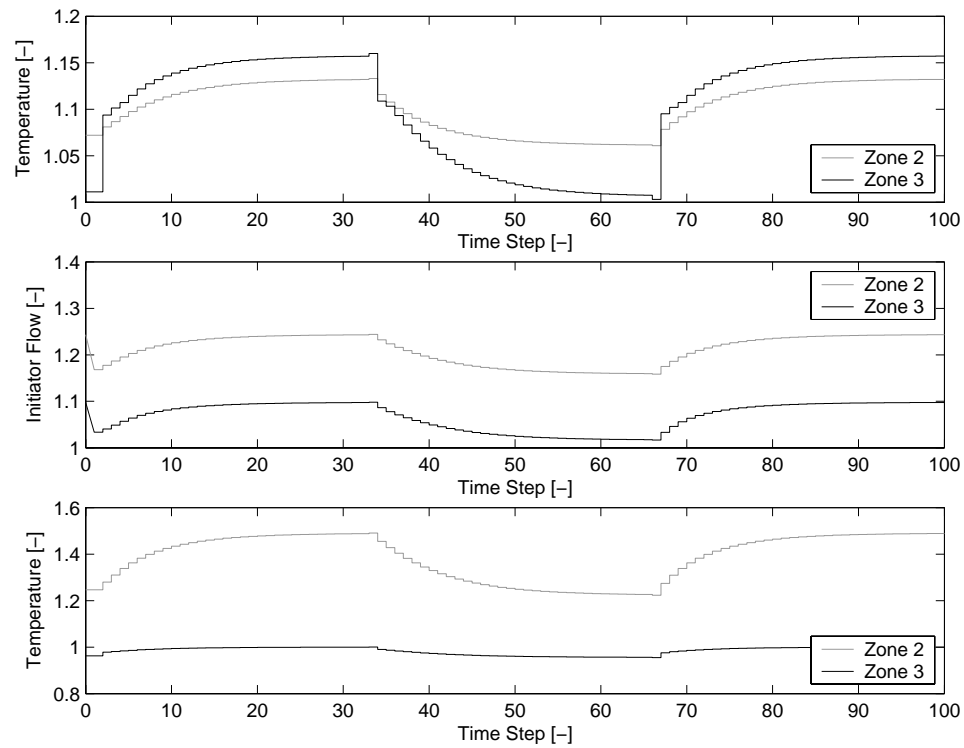


Figure 8.16: NMPC controller behavior. Jacket inlet temperatures (top). Initiator flows (middle). Side-streams temperatures (bottom).

objective to the NMPC controller. This economics-oriented controller can be seen as an *all-at-once* RTO-NMPC strategy (D-RTO) that directly optimizes the process performance. Such an strategy can account for the dynamic disturbances that affect the process profitability and can exploit more efficiently the degrees of freedom available in distributed LDPE reactors [118].

For the design of the D-RTO or economics-oriented NMPC controller, we will incorporate a measure of the process profitability on the controller objective. In this case, we have decided to incorporate the accumulated production rate. This is justified by the fact that in high-throughput LDPE processes the operating costs are largely affected by production losses associated with fouling. The objective function of the controller will take the following form,

$$\begin{aligned} \Phi^k = & -\alpha_e \int_0^{N\delta} Production(t)dt \\ & +\alpha_t \int_0^{N\delta} \int_0^{x_L} [(\gamma(t,x) - \bar{\gamma}(t,x))^T \mathbf{Q}^{-1}(\gamma(t,x) - \bar{\gamma}(t,x))] dxdt \end{aligned}$$

Here, the controller tries to minimize the transition time from the current state to the targets and simultaneously maximizes the reactor production. We have found that a *purely economic objective* leads to *ill-posed* optimization formulations due to the large number of degrees of freedom encountered in LDPE reactors and due to the distributed nature of these systems. With a purely economic objective we cannot guarantee the uniqueness of the solution (e.g. SSOC does not hold) and this also makes the NMPC problems extremely difficult to solve. These observations agree with those of Huesman and coworkers [61] and with those of Skogestad in the context of steady-state RTO [113]. In this work, we add a tracking term to *regularize* the problem and thus obtain smoother solutions. Notice the addition of positive weighting factors α_t and α_e , which are controller design parameters. In the context of LDPE processes, the tuning factor α_t reflects how much is the controller allowed to deviate from the reference temperature profile in order to maximize the production rate. Note also that the production term is a highly complex nonlinear mapping of the

model states. Here, we emphasize on the importance of using exact second-order derivative information to handle these types of objectives.

We have tested the D-RTO controller using the same fouling-defouling scenario analyzed in the previous section. In Figure 8.17 we compare the performance of the NMPC (tracking) and the D-RTO (economic) controllers. In the top graph we see that the tracking controller keeps the reactor temperature profile close to the given reference. On the other hand, in the middle graph we see that D-RTO controller responses tend to deviate from the reference. In the bottom graph we see that the adjustment of the temperature profile results in a direct increase on the reactor production levels of more than 10% during both the fouling and the defouling stages. These results are quite promising since, as we have mentioned, a slight increase in production in high-throughput LDPE processes can represent millions of dollars in annual savings. The D-RTO controller recognizes that the given reference profile *is not the optimum profile* in terms of reactor productivity and it tends to correct it. Since the controller also manipulates the chain-transfer agent flow rate, these changes can be made without having to sacrifice polymer quality (e.g. melt index). Note that the D-RTO controller increases the peak temperature levels in the third and fourth zone and it tends to decrease their corresponding inlet temperatures. On the other hand, note that the controller leaves the temperature profile of the first two zones almost unchanged. This implies that *production bottlenecks* are mostly associated with the third and fourth zones. To illustrate this in more detail, in Figure 8.17 we present the relative production improvements for the first zones. Note that the improvements are most notable downstream of the reactor.

In general, we can conclude that the main benefit of the D-RTO controller is the ability to find better strategies to distribute the polymer production across the multiple zones. Since this in turn depends on the time-varying fouling levels at each zone, the controller is an efficient alternative to manage fouling issues. Nevertheless, from the bottom graph of Figure 8.17 we can also see that the D-RTO controller *still needs to give away some production* during the fouling period. This implies that the reactor performance is still

affected by fouling.

8.4 Moving Horizon Estimation

As we have seen, there exist multiple sources of time-varying uncertainty arising in LDPE processes that are difficult to predict through the reactor model. In particular, we have seen that the fouling onset is difficult to predict through mechanistic models. Nevertheless, the availability of a first-principles model allows to map the sources of uncertainty to parameters with physical meaning that can be estimated on-line. A second issue arising in industrial LDPE units is that the available on-line measurement information is rather limited. Consequently, it is difficult to analyze and understand *internal* phenomena taking place at the reactor core. In this section, we couple the first-principles dynamic model to a MHE state estimation strategy to overcome these limitations. The estimated states and time-varying parameters will be later used to provide output feedback through the NMPC controller.

8.4.1 MHE Problem Formulation

Consider the scenario in which the LDPE process is currently located at sampling time t_k and we have a *past* measurement history $\{\eta_k, \eta_{k-1}, \dots, \eta_{k-N}\}$ distributed over a time horizon containing N sampling times of equal length $\delta = t_k - t_{k-1}$. Using this information, we seek to *infer* the current state of the reactor $z^k(x), w^k(x), y^k(x)$ and the parameters p^k through the rigorous model. We assume that the model structure is *correct* and that all the uncertainty related to the real process can be encapsulated in the model parameters $p(t)$, in the initial conditions $z^0(x)$ and in the measurement errors. Following this reasoning, an estimate of state of the process can be computed through the solution of an estimation problem of the form,

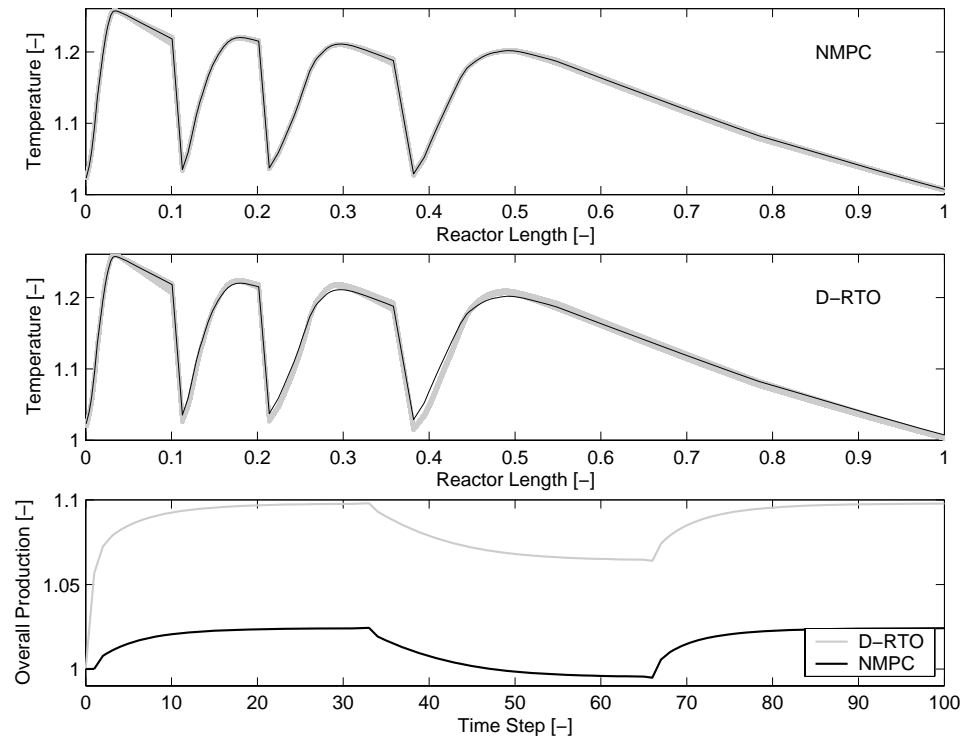


Figure 8.17: Comparison of NMPC and D-RTO controller responses. Closed-loop response of temperature profile for NMPC (top) - dark line is set-point and gray lines are the controller responses. Closed-loop response of temperature profile for D-RTO (middle) - dark line is set-point and gray lines are the controller responses. Reactor overall production responses (bottom).

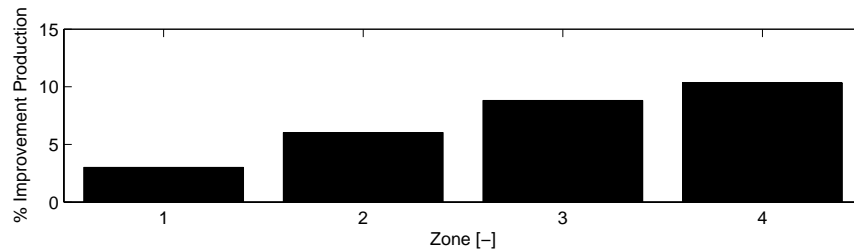


Figure 8.18: Relative production gains for individual reactor zones with D-RTO.

$$\begin{aligned} \min_{p(t), z^0(x), u(t)} \quad & (z^0(x) - \bar{z}^0(x))^T \mathbf{\Pi}_0^{-1} (z^0(x) - \bar{z}^0(x)) \\ & + \sum_{k=1}^N \sum_{i=1}^{N_m} (\eta_{k-N}^{(i)} - \gamma(j\delta, x_i))^T \mathbf{R}^{-1} (\eta_{k-N}^{(i)} - \gamma(j\delta, x_i)) \end{aligned} \quad (8.9a)$$

s.t.

$$\frac{\partial z}{\partial t} + \nu(t, x) \frac{\partial z}{\partial x} + \kappa(t, x) \frac{\partial^2 z}{\partial x^2} = f_z(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.9b)$$

$$\frac{\partial w}{\partial x} = f_w(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.9c)$$

$$0 = f_y(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.9d)$$

$$\gamma(t, x) = \chi(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.9e)$$

$$0 \geq g(z(t, x), w(t, x), y(t, x), p(t), u(t)) \quad (8.9f)$$

$$z(0, x) = z^0(x) \quad (8.9g)$$

$$0 = \varphi \left(z(t, 0), z(t, x^L), \frac{\partial z}{\partial x}(t, 0), \frac{\partial z}{\partial x}(t, x^L), w(t, 0), u(t) \right) \quad (8.9h)$$

$$t \in [0 \ N\delta].$$

Note that the process measurements are only defined at discrete points in time $j\delta$ and space x_i . Symbol $\eta_k^{(i)}$ denotes the i -th spatial measurement in vector η_k . The total number of measurements in space is denoted by N_m . The measurement vector contains both output and input measurements with error covariance $\mathbf{R} \in \Re^{n_\gamma \times n_\gamma}$. The MHE problem contains a large number of degrees of freedom including the parameters, the initial conditions for the dynamic states and the inputs. Variable $\bar{z}^0(x)$ denotes the *a priori* value of the initial state with covariance $\mathbf{\Pi}_0 \in \Re^{n_z \times n_z}$. The first term in the objective function is the *arrival cost*, which summarizes past measurement information before sampling time t_{k-N} .

From the solution of the optimization problem, we extract the estimate of the current state of the process as $\hat{z}^k(x) \leftarrow z^*(\delta N, x)$, $\hat{w}^k(x) \leftarrow w^*(\delta N, x)$, $\hat{y}^k(x) \leftarrow y^*(\delta N, x)$, the current parameters $\hat{p}^k \leftarrow p^*(\delta N)$ and the reconciled inputs. At the next sampling time t_{k+1} , we drop the last measurement and incorporate the new one to obtain the new measurement history $\{\eta_{k+1}, \eta_k, \dots, \eta_{k+1-N}\}$. Accordingly, we update the *a priori* estimate of the initial state using the previous solution as $\bar{z}^0(x) \leftarrow z^*(\delta, x)$. In some cases, it is also necessary to update the covariance matrix $\mathbf{\Pi}_0$ using the covariance of the predicted state $z^*(\delta, x)$ to

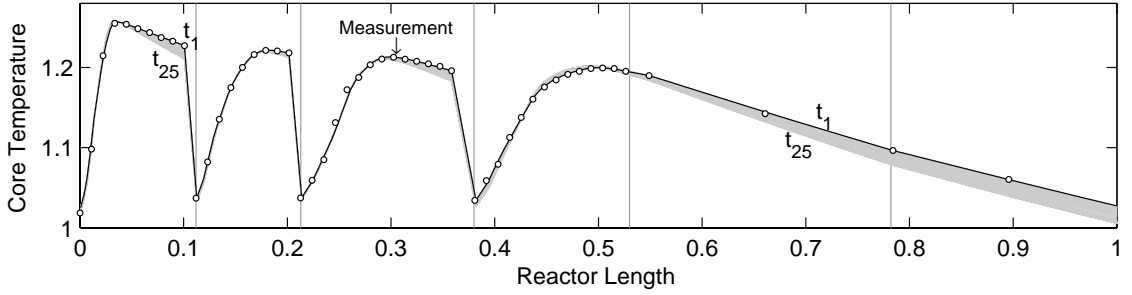


Figure 8.19: Closed-loop response of reactor core temperature profile and position of thermocouples.

account for poor estimates of the initial conditions. To solve the MHE problem, we use the same model implementation and solution approach of the NMPC problem. We use IPOPT to solve the NLPs using exact derivative information coming from AMPL. In addition, we extract the covariance matrix $\mathbf{\Pi}_N$ from problem (8.9) for the current state estimates $\hat{z}^k(x)$ using the reduced Hessian extraction capabilities described in Chapters 3 and 6.

8.4.2 Case Studies

We analyze the performance of the MHE estimator in the closed-loop fouling-defouling scenario discussed in the previous section. We obtain the measurement information η_k from the closed-loop NMPC simulation. In Figure 8.19, we present the closed-loop response of the temperature profile during the first defouling cycle (first 25 time steps). In addition, we indicate the specific location of the thermocouples measuring the reactor core temperature. In Figure 8.20 we illustrate the dynamic response of the jacket and we present the position of the available jacket thermocouples.

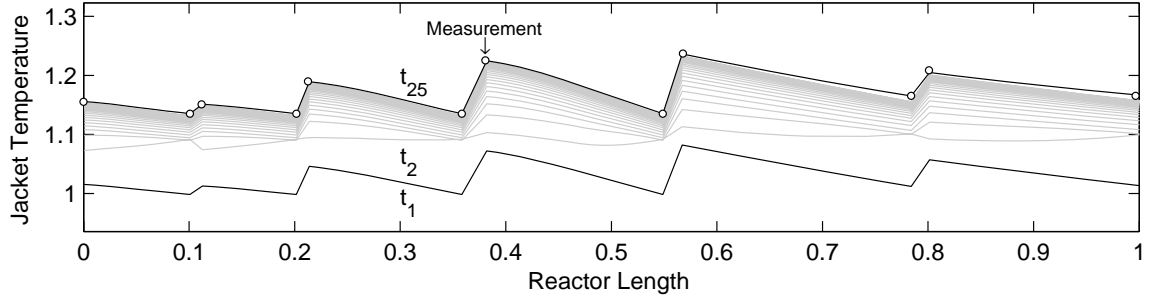


Figure 8.20: Closed-loop response of jacket cooling water and position of thermocouples.

Nominal Case

The NMPC simulation results provide the dynamic profiles for all the reactor states that we use as the *true* states to test the estimator. To analyze the *nominal* performance of the estimator, we consider the case in which we have a wrong *a priori* guess of the wall and jacket profiles and heat transfer coefficients at time t_0 . In addition, we assume that no measurement noise is present. We use the initial guess and the available measurement information to reconstruct the dynamic profiles for the heat transfer coefficients and all the state profiles of the process.

We consider two scenarios. In scenario 1, we generate the initial guess for the wall and jacket temperature profiles by perturbing the *true* profiles by -10%. In absolute values, this corresponds to a perturbation of approximately -20°C to a reference value of 40°C (i.e. our guess of the profile is underestimated 20°C). In scenario 2, we perturb the initial guess of the wall and jacket temperature profile by a Gaussian zero-mean disturbance with a standard deviation of 20°C . In both scenarios, we also perturb the initial guess of the heat transfer coefficients by +50 %. The results are presented in Figures 8.21 and 8.22. In the top graph of Figure 8.21 we can observe that the estimator is able to converge to the true value of the heat transfer coefficients. In the middle graph, note that the initial guess of the wall temperature profile in scenario 1 is shifted and the estimator is able to converge

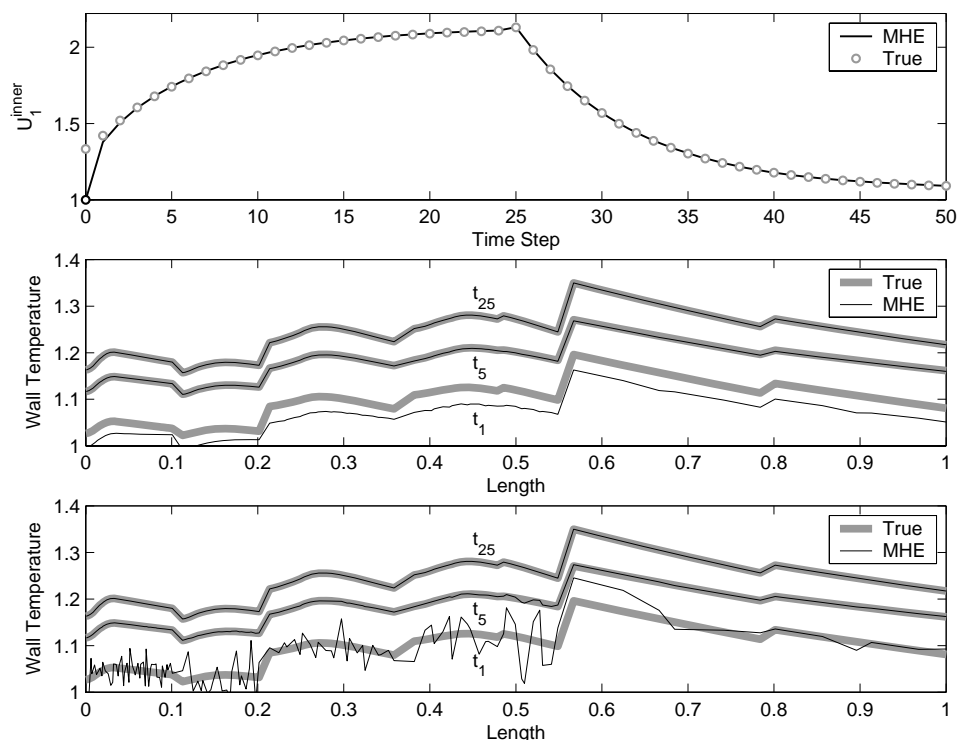


Figure 8.21: Convergence properties of MHE estimator for perturbations on initial guess. Convergence to heat transfer coefficient of first reaction zones in scenario 1 (top). Convergence to wall temperature profile in scenario 1 (middle) and scenario 2 (bottom).

to the true wall temperature profile at sampling time t_5 . In the bottom graph, we note the large disturbances added to the random initial guess of the wall profile in scenario 2. The estimator is also able to reconstruct the true wall temperature profile at around time t_5 .

In the top graph of Figure 8.22 we demonstrate that the estimator also converges to the true jacket temperature profile. In the middle graph, we can observe that the available temperature measurements provide enough information to reconstruct the concentration of radicals across the reactor at the first sampling time t_1 . Here, it is also possible to identify the addition of initiator flow rates at the beginning of each reaction zone and the fast consumption. In the bottom graph of Figure 8.22 we see that the temperature profile

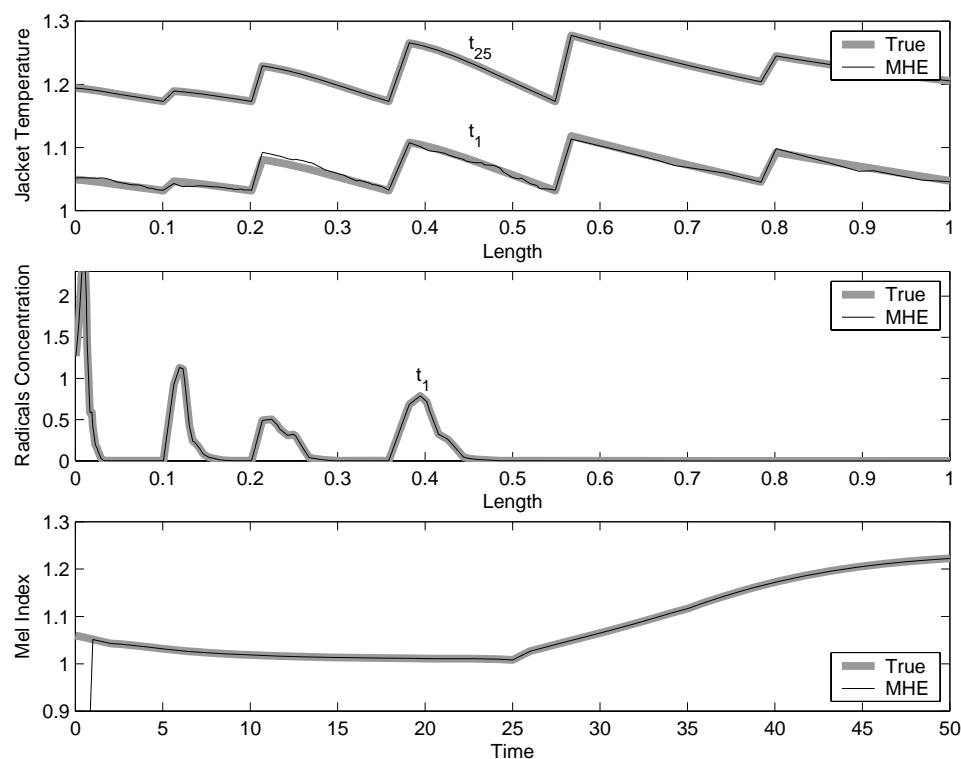


Figure 8.22: Convergence properties of MHE estimator for perturbations on initial guess. Convergence to jacket temperature profile in scenario 1 (top). Convergence to radicals concentration profile at time step t_1 (middle). Convergence to polymer melt index at the reactor outlet (bottom).

provides enough information to infer quickly the polymer melt index at reactor outlet. These results are quite surprising as in many processes temperature measurements are not sufficient to infer all the model states. However, as we have seen in the off-line parameter estimation case study, it seems that the *shape* of the temperature profiles arising in LDPE reactors presents a significant degree of *excitation* that tends to make the measurement data highly informative.

The solution of all the MHE problems considered satisfy second order conditions (SSOC)

indicating that the process is locally observable using the available measurement information. To quantify the observability properties of the reactor, we perform a numerical analysis of the covariance matrix of the estimated wall temperature profile ($\mathbf{\Pi}_N$) for different values of the estimation horizon N . The results are presented in Figure 8.23. In the top graph, we plot the diagonal elements of the covariance matrix corresponding to each wall temperature along the reactor. Note that the covariance levels decay quickly as we add measurements into the horizon. In addition, the covariance profile becomes similar in shape to that of the temperature profile (e.g. where the measurements are present). It is also clear that the wall temperature can be inferred more reliably close to the reactor inlet than downstream the reactor. In the bottom graph, we plot the maximum eigenvalue of the covariance matrix $\lambda_{max}(\mathbf{\Pi}_N)$ as a function of the estimation horizon. We compare three cases. The first case involves the estimator using the full measurement information available (core and jacket temperatures, inputs and conversion). The profile is presented as a solid gray line. Note that the maximum eigenvalue decays rather quickly and reaches the steady-state covariance $\mathbf{\Pi}_\infty$ as the estimation horizon length is increased. A few time steps seem to be sufficient to obtain reliable estimates of the wall profile. In the second case, we eliminate the conversion measurement from the objective function (dashed black line). Note that the covariance of the wall profile is not affected by this modification. This implies that the conversion measurement is redundant (conversion can be inferred directly from the temperature profile). In the third case, we eliminate one third of the reactor core measurements distributed along the reactor. Note that this change strongly affects the covariance levels of the estimated states. This illustrates the importance of the temperature profile on the observability properties of the reactor.

Output Noise Filtering

We demonstrate the performance of the estimator for cases in which random Gaussian noise is added to the temperature measurements. In the first scenario, we add noise with a

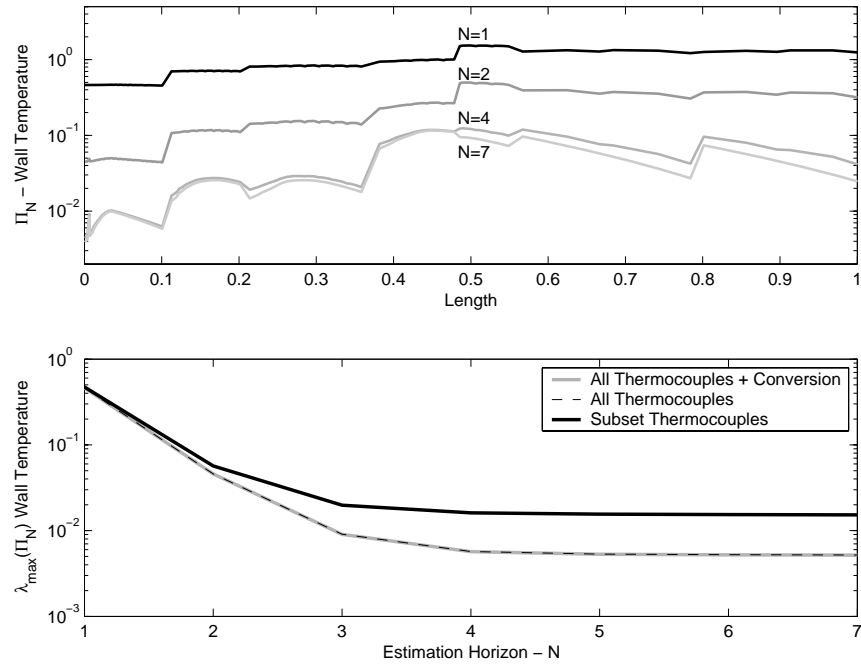


Figure 8.23: Effect of estimation horizon length on convergence of estimator. Covariance of wall temperature profiles for different horizon lengths N (top). Decay of maximum eigenvalue of covariance matrix for different measurement scenarios (bottom).

standard deviation $\sigma = \pm 3^\circ\text{C}$ to each thermocouple measurement. This is a fairly standard noise level encountered in industrial reactors. In the second scenario we increase the level of noise to $\pm 5^\circ\text{C}$. In both scenarios, the corresponding diagonal elements of the covariance matrix \mathbf{R} are set to $\frac{1}{\sigma^2}$. The results are presented in Figure 8.24. As can be seen, the estimator is able to reject the measurement noise and reconstruct the profile of the heat transfer coefficients.

Input Noise Filtering

As we have seen in EVM formulations for parameter estimation, input measurement noise leads to biased state estimates. To demonstrate this, we consider the scenario in which the inlet temperature of the cooling water of all the zones is contaminated with Gaussian noise with $\sigma = \pm 3^\circ\text{C}$. The results are presented in Figure 8.25. In the top graph, we illustrate the inlet cooling water temperature corrupted with noise. In the middle graph, we can see that the input noise disrupts the convergence of the estimator to the true jacket temperature profile at time t_{50} (top graph). The estimate is biased. In the bottom graph, we observe that input noise also degrades the convergence of the estimator to the true heat transfer coefficients.

In order to filter out the input noise, the values of the input variables are reconciled simultaneously in the MHE formulation. For this EVM-MHE formulation, we define all these inputs as degrees of freedom in the NLP. In addition, we incorporate the corresponding least-squares terms to the objective function and modify the corresponding diagonal elements of the covariance matrix \mathbf{R} . As in the previous standard MHE case study, we corrupt the measurements of the inlet cooling water temperatures with Gaussian noise with $\sigma = \pm 3^\circ\text{C}$. In the top graph of Figure 8.26, we can see that the estimator is able to filter out the input noise and infer the true input. In the middle graph, we see the resulting smoother convergence of the heat transfer coefficients to the true value, compared to those observed in Figure 8.25. In the bottom graph, we see the bias elimination on the jacket temperature

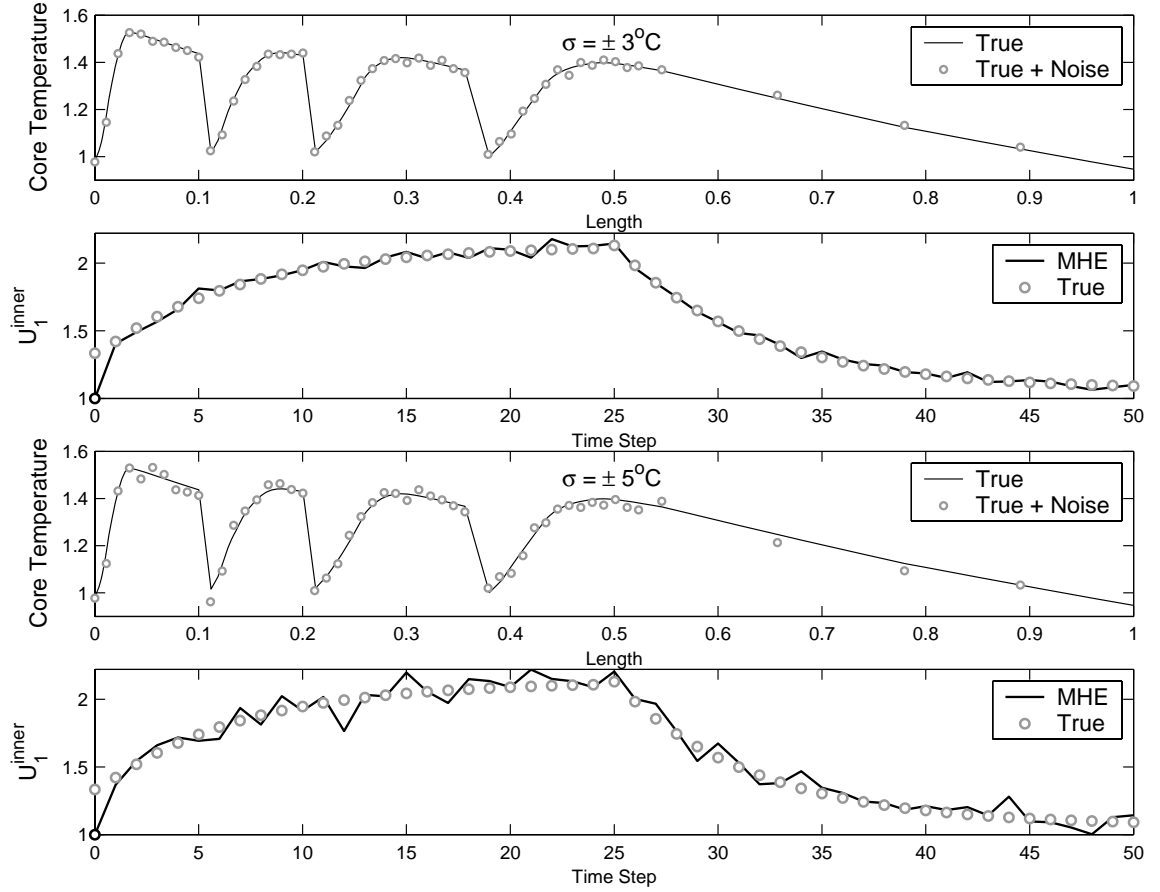


Figure 8.24: Effect of output noise on performance of estimator. Output noise on reactor core temperature with $\sigma = \pm 3^\circ \text{C}$ and convergence to heat transfer coefficients (top). Output noise with $\sigma = \pm 5^\circ \text{C}$ and corresponding heat transfer coefficients (bottom).

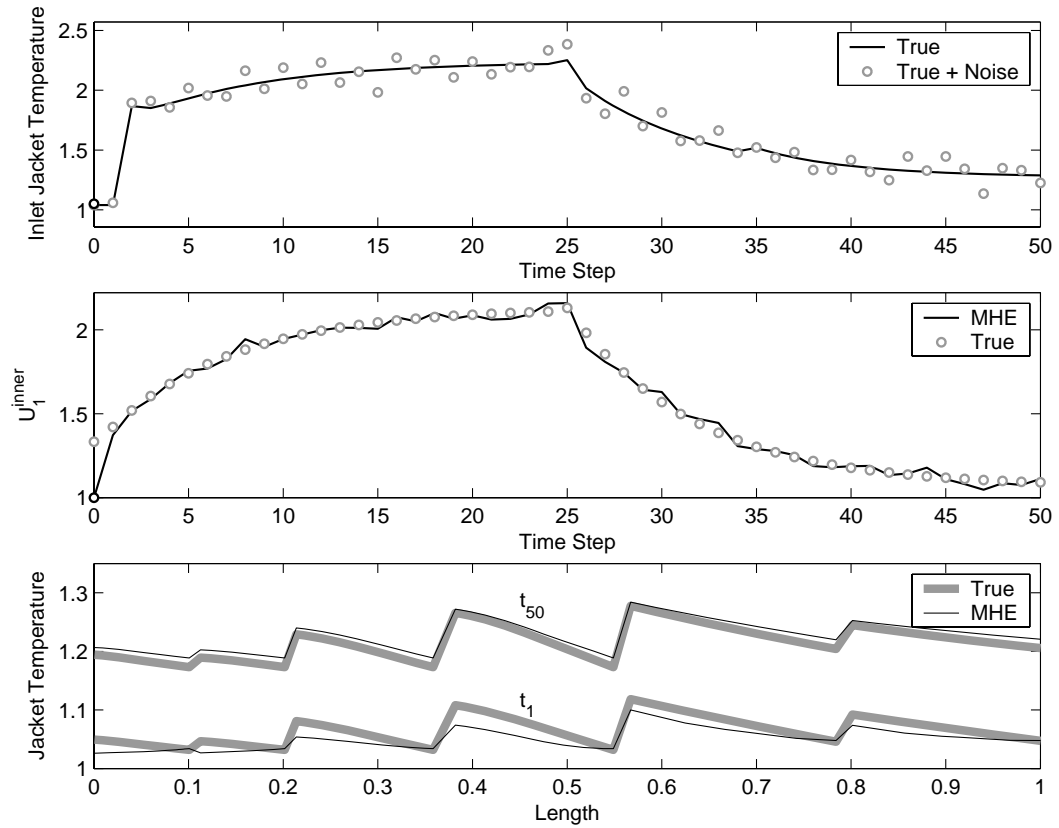


Figure 8.25: Effect of *unrejected* input noise on performance of estimator. True and noisy jacket inlet temperatures for Zone 1 (top). Estimated heat transfer coefficient for Zone 1 (middle). Biased estimates of wall temperature profile (bottom).

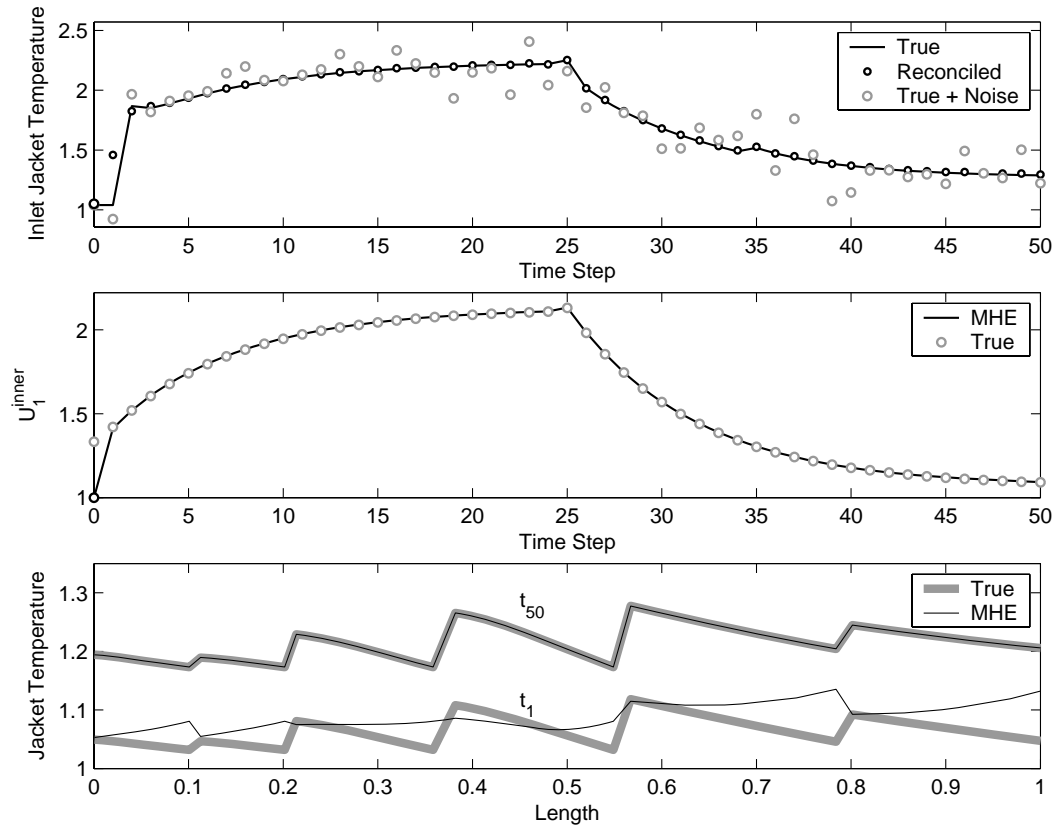


Figure 8.26: Effect of *rejected* input noise on performance of estimator. True, reconciled and noisy jacket inlet temperatures for Zone 1 (top). Estimated heat transfer coefficient for Zone 1 (middle). Unbiased estimates of wall temperature profile (bottom).

profile through the EVM-MHE formulation.

8.5 NMPC-MHE Coupling

In the previous sections we have evaluated the performance of NMPC and MHE strategies for LDPE processes and we have discussed their potential benefits. In this section, we couple these strategies and analyze their performance in an output feedback scenario. In addition, we discuss computational scale-up issues, and use advanced-step strategies to overcome computational delays.

8.5.1 Case Studies

We simulate an output feedback scenario in which the reactor is fouled and defouled over time. Again, this is done by ramping the reactor heat-transfer coefficients (HTCs). Since the LDPE reactor model cannot predict the fouling disturbance, we use the MHE estimator to estimate the HTCs p^k and the unmeasured model states $\tilde{z}_k(x)$ (e.g. wall temperature profile) at each time step. The objective of the NMPC controller is to use the estimated reactor state $\tilde{z}_k(x)$ to drive the reactor outputs to the desired target by computing the optimal control action $u_k = h(\tilde{z}_k(x))$. In this simulated scenario, we generate the true plant response $z^k(x)$ from the model with the *true* HTCs. Since the MHE estimator starts with wrong guesses of the state and the parameters, and since we add Gaussian measurement noise, it will introduce an estimation error that acts as a disturbance to the plant. In addition, note that since the fouling phenomenon cannot be predicted, the estimator can only converge to the true value of the HTCs one step behind. Once the HTC disturbance vanishes, the estimator converges to the true values and the NMPC controller recovers its nominal stability properties.

In the top graph of Figure 8.27 we compare the predicted temperature profile of the

NMPC controller at time step 25 and the corresponding profile corrupted with $\sigma = 3\%$ Gaussian noise. In the middle graph we see that the MHE estimation converges to the true profile of wall in less than 10 time steps despite of the noise and the wrong initial guess. In the bottom graph we see the effect of noise on the control actions by the NMPC controller. In the top graph of Figure 8.28 we illustrate the convergence of the MHE estimator to the jacket water temperature at a particular point along the reactor. In the middle graph we illustrate the convergence of wall temperature at the same axial position. In the bottom graph we can see that the NMPC controller is able to stabilize the system despite of the multiple disturbances and is able to keep the reactor temperature profile close to the target. It is important to emphasize that the strong observability properties of the LDPE reactor result in good robustness and stability properties of the NMPC controller. In particular, note that the number and location of the temperature measurements available along the reactor are particularly important for this.

8.5.2 Computational Issues and Advanced-Step Strategies

In this section, we discuss computational issues associated with the proposed MHE and NMPC strategies for LDPE reactors. In all our computational studies, we used the full discretization approach together with IPOPT to solve both NLPs. We initialized tested the MA27 linear solver to factorize the KKT matrix. We found similar results to those discussed in the off-line parameter estimation problem. MA27 is in general robust and can handle the poor scaling of the KKT matrix resulting from the LDPE model quite efficiently. However, the minimum degree ordering strategy used in MA27 tends to introduce large amounts of fill-in and the NMPC and MHE problems become expensive to solve even with short time horizons. Motivated by this, we performed testing with the MA57 linear solver. We found that MA57 is in general is much faster than MA27 and can handle fill-in issues more efficiently. Nevertheless, we have also found that choosing an appropriate preordering strategy in MA57 is critical to obtain fast factorizations [52].

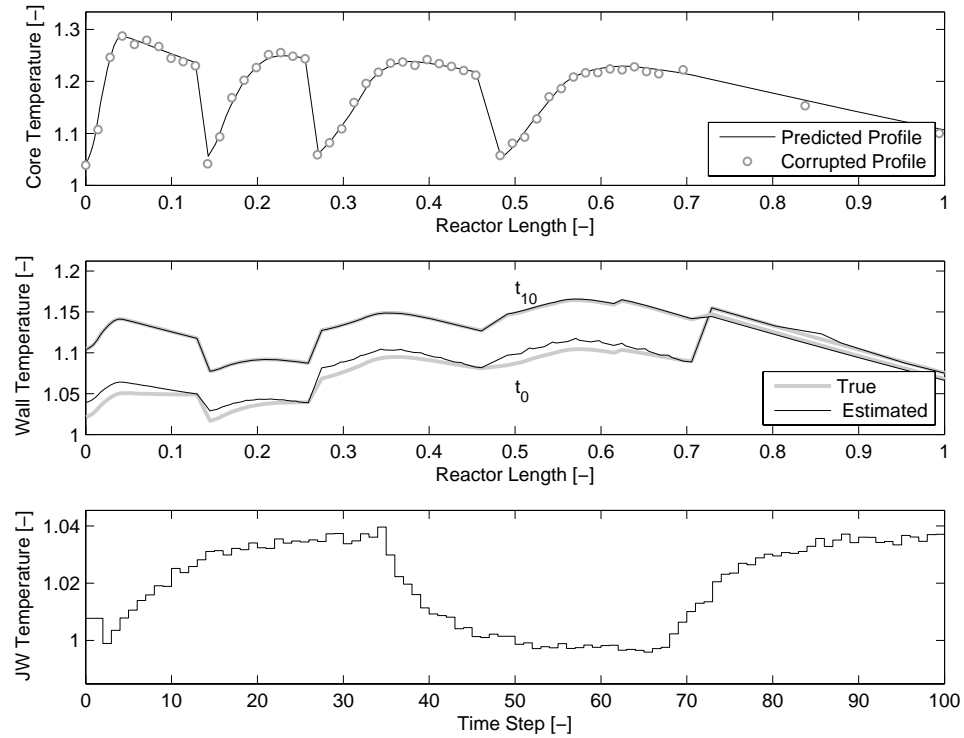


Figure 8.27: Performance of coupled NMPC-MHE for output feedback. Predicted temperature profile at time step 25 and corresponding profile corrupted with Gaussian noise (top). Convergence of MHE estimator to true wall profile (middle). Jacket water inlet temperature of Zone 2 computed by NMPC controller (bottom).

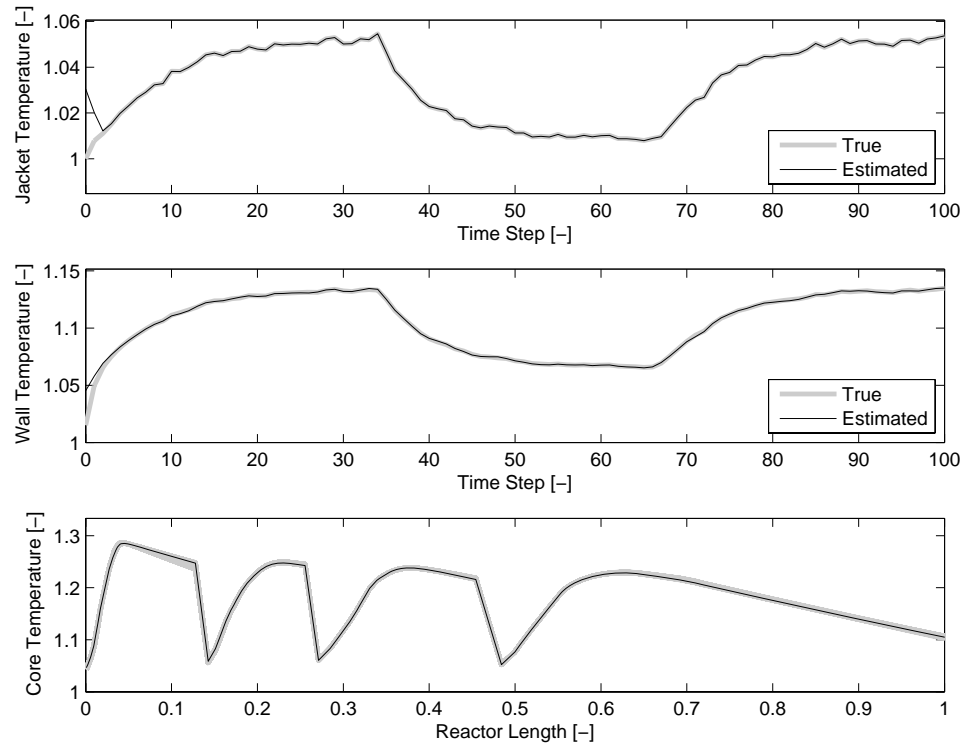


Figure 8.28: Performance of coupled NMPC-MHE for output feedback. Convergence of MHE estimator to jacket temperature at a particular location of the reactor (top). Convergence of MHE estimator to wall temperature (middle). Closed-loop responses of NMPC for temperature profile (bottom).

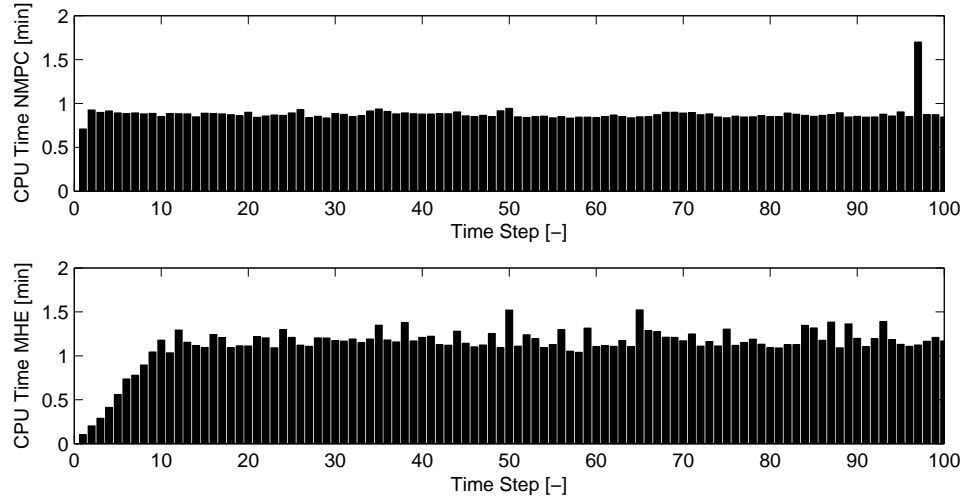


Figure 8.29: Solution times for NMPC (top) and MHE problems (bottom) with horizons of $N = 10$ time steps.

In order to motivate the discussion, we present results on the total wall-clock time required to solve the tracking NMPC and the MHE problems in the output feedback case study of the previous section. All calculations were obtained using a quad-core Intel processor running Linux at 2.4 GHz. We use shifting strategies to warm-start subsequent NMPC and MHE problems and we set the barrier parameter of IPOPT to $\mu = 1 \times 10^{-6}$. The solution times also include some overhead coming from I/O communication tasks arising in the implementation and from AMPL, which requires some time to generate the derivative information before calling the NLP solver. The prediction and estimation horizons N were set to 10 time steps (20 minutes) and sampling times of 2 minutes were used. The NMPC problem consists of an NLP with 80,950 constraints and 370 degrees of freedom. The MHE problem consists of an NLP with 80,300 constraints and 648 degrees of freedom. As can be seen in the top graph of Figure 8.29, the overall solution time for the NMPC problem is around 60 seconds. The NLP solver requires 3-4 iterations to converge the problems. In the bottom graph we present total solution times for the MHE estimator. In this case, the estimator is initialized in batch mode (accumulate measurements until an estimator horizon

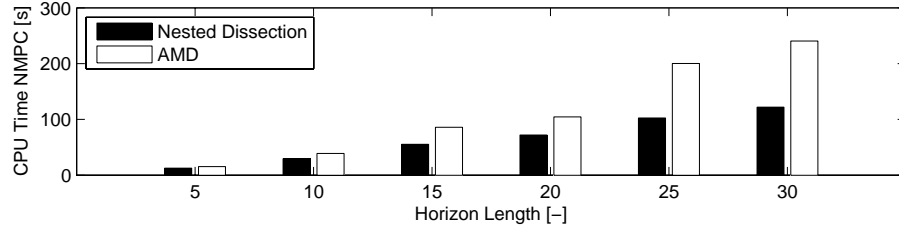


Figure 8.30: Scale-up results for solution times of NMPC problem with different horizon lengths.

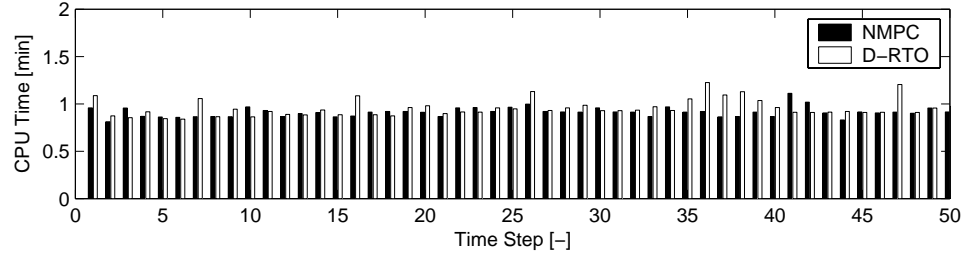


Figure 8.31: Comparison of solution times for NMPC problem with tracking and economic objectives.

of N time steps is filled). Once the estimation horizon is complete, IPOPT takes around 70 seconds to solve the problem (3-4 iterations are required).

In Figure 8.30, we present scale-up results of the solution time for the NMPC problem with increasing horizon length. We compare the impact of an approximate minimum degree ordering (AMD) strategy and nested dissection strategy reordering on the solution time of the NMPC problem (without refinement or overhead). Note that the multi-level nested dissection strategy is more efficient here and achieves a linear scale up. Using this strategy, a NMPC problem with $N = 30$ time steps resulting in an NLP with 242,850 constraints and 1,110 degrees of freedom is solved in around 2 minutes. The AMD strategy shows quadratic scale-up and the largest problem requires 4 minutes. This huge difference can be attributed

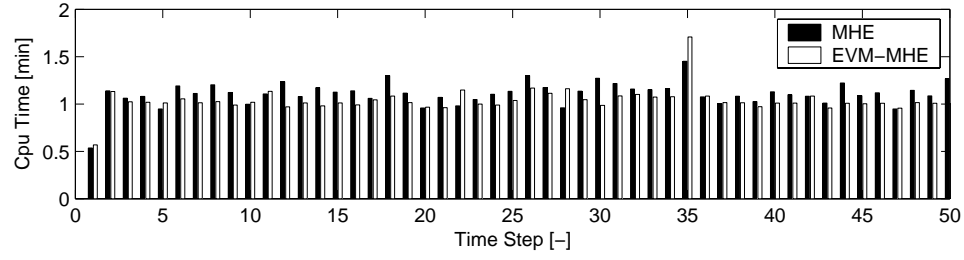


Figure 8.32: Comparison of solution times for MHE and EVM-MHE problems.

to the fact that the nested dissection algorithm can identify coarse-grained structures present in the NMPC problem more easily (LDPE multi-zone model, finite element structure, etc.), while AMD tends to focus on fine-grained structures. Similar results have been found for the MHE problem. An MHE problem with an estimation horizon of 30 time steps results in an NLP with 244260 constraints and 1878 degrees of freedom. This problem can be solved with a nested dissection strategy in around 2.5 minutes.

In Figure 8.31 we illustrate the solution times for NMPC problems with tracking and economic objectives. We can see that if exact derivative information is used to solve the problems, the solution times (e.g. factorization time and number of iterations) are not altered by changes in the objective function. In Figure 8.32 we compare the solution times of MHE and EVM-MHE problems in a particular case study. In general, we have found that the use of exact derivative information allows to solve both problems in the same number of iterations [99]. However, we have also found that if the default AMD preordering strategy of MA57 is used, the factorization of the KKT matrix arising from the EVM problems becomes extremely expensive [128]. Nevertheless, in Figure 8.32 we can see that, when a nested dissection strategy is used, the factorizations and solution times of both the MHE and the EVM-MHE problems are very similar.

Advanced-Step Strategies

From the previous results we can conclude that a full-discretization approach coupled with an sparsity-exploiting NLP solver results in a highly efficient computational strategy to solve large-scale NMPC and MHE problems. In particular, we have demonstrated that this approach scales well with problem size and number of degrees of freedom. Here we emphasize that the development of new and powerful linear solvers plays a crucial role in achieving this favorable complexity. Nevertheless, from the previous results it also becomes clear that *even if we have a fast strategy to solve the problems, the solution time will always become a bottleneck as we consider larger and larger applications*. For instance, in the context of the output feedback LDPE case study, the time required to solve the MHE problem to obtain the state estimate plus the time required to solve the NMPC problem to obtain the control actions is more than 2 minutes. This solution time is in fact longer than the assumed sampling time. This problem becomes more obvious as we consider longer time horizons. Here, we demonstrate that these limitations can be avoided with advanced-step strategies.

We demonstrate the performance of a coupled advanced-step NMPC-MHE strategy on the same output feedback scenario presented in the previous section. Again, a prediction horizon N of 10 time steps (20 minutes) and sampling times of 2 minutes have been used. Here, we demonstrate that the approximation errors introduced by NLP sensitivity are negligible and do not destabilize the controller. In this scenario, we recognize that since the plant response differs from that of the NMPC controller prediction and we introduce noise, the asMHE estimator will see a difference between the measured and the predicted outputs and will correct on-line using NLP sensitivity. We have found that the approximation errors are negligible and the asMHE estimator has almost identical convergence properties to that of the ideal MHE estimator. In the top graph of Figure 8.33, we can see that the asMHE estimates are identical to those of the ideal or optimal MHE estimator. Using the estimated states and HTCs, the asNMPC controller then corrects the predicted state on-line. In the middle graph of Figure 8.33 we present the closed-loop response of one of the jacket water

inlet temperatures for the asNMPC controller and of its ideal NMPC counterpart. As can be seen, both control actions are identical. In the bottom graph of this figure we can see that the asNMPC controller is able to stabilize the temperature profile around the target. *The sensitivity calculations for both the NMPC and MHE problems require less than 0.1 seconds.*

The total background times required to converge the predicted MHE and NMPC problems and update the KKT matrices are very similar to those presented in Figure 8.29. However, we have found that the NLP sensitivity approximations for the Lagrange multipliers tend to deviate from the optimal ones. This is mainly associated with the ill-conditioning of the LDPE model that generates extremely large values for some of the multipliers (e.g. 1×10^7). When the NLP sensitivity approximations are used to warm-start the background problems, the NLP solver requires many more iterations to converge them. In order to avoid this, we perform an on-line *refinement* of the multipliers through fixed-point iterations. We have found that around 5-10 fixed-point suffice to *fully converge* the NMPC and MHE problems on-line. The time required by 10 fixed-point iterations is around 1 second. These results are interesting because this implies that the proposed advanced-step strategies can also be seen as highly effective warm-start strategies. The predicted problems provide the fixed KKT matrices and the problems can be *solved on-line to optimality* using fast fixed point iterations. In addition, note that the advanced-step strategy allows to decouple the solution of the background MHE and NMPC problems. In the case study analyzed, this allows to keep the sampling time to 2 minutes. With this, we can conclude that the proposed advanced-step NMPC-MHE strategies allow to consider implementations with long time horizons, reasonable sampling times and negligible feedback delays.

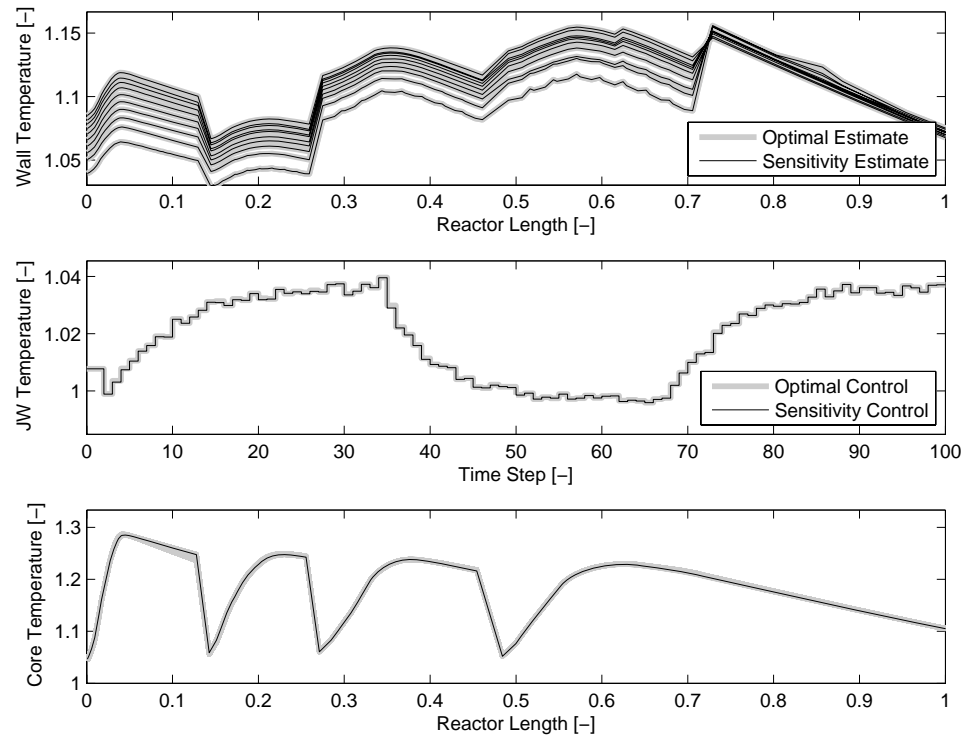


Figure 8.33: Effect of NLP sensitivity errors on performance of advanced-step MHE and NMPC strategies in LDPE case study.

8.6 Concluding Remarks

We have discussed operational problems arising in industrial low-density polyethylene (LDPE) tubular reactors. In particular, we have seen that persistent dynamic disturbances such as fouling strongly affect the profitability of the process. In addition, we have seen that the distributed nature of LDPE reactors gives rise to complicated operating policies. As a consequence, there exists a strong motivation to incorporate sophisticated first-principles models to dictate the operation of these complex processes systematically.

We present results on a large-scale parameter estimation study in which we tune a LDPE reactor model using industrial measurement data. We propose an simultaneous (all-at-once) strategy for the solution of parameter estimation problems incorporating a detailed first-principles DAE model and multiple data sets. The estimated parameters result in significant improvements on the model predictions. In addition, a systematic estimation analysis allows to determine which parameters can be estimated reliably and reveals important structural model deficiencies. Finally, we demonstrate that a Schur decomposition strategy is effective in solving very large-scale estimation problems with multiple data sets.

We have incorporated a dynamic version of the reactor model in a NMPC strategy. Using a fouling-defouling scenario, we demonstrate that a tracking NMPC controller is able to stabilize the reactor in the face of persistent fouling disturbances while keeping the polymer properties at target. We also demonstrate that if an economic objective function is added, the NMPC controller can exploit the multiple degrees of freedom of the process to simultaneously optimize the process profitability. In particular, we have found that an economics-oriented controller can improve the production rate by more than 10%.

We presented a MHE strategy to infer time-varying fouling behavior and to provide estimates of the plant states to the NMPC controller. A numerical analysis of the covariance matrix of the wall temperature reveals that the multiple temperature measurements available in the reactor make the system strongly observable. The measurements help the MHE

estimator to converge quickly to the true model states even in the face of poor initial guesses. In addition, we demonstrate that the MHE estimator can reject both input and output noise effectively. We have coupled the MHE estimator to the NMPC controller to provide output feedback to the LDPE process. We have found that the fast convergence properties of the MHE estimator aid the NMPC controller in stabilizing the process effectively.

We have demonstrated that a full-discretization approach coupled to a full-space interior point solver results in an fast strategy to solve both the NMPC and MHE problems. In addition, we have demonstrated that this approach scales linearly and, at most quadratically, with problem size. We have seen that the use of an efficient linear solver is crucial in achieving this. Finally, we implement advanced-step NMPC-MHE output feedback strategy. We demonstrate that a highly sophisticated dynamic model can be accommodated on-line to provide nearly instantaneous feedback to the LDPE process with negligible approximation errors.

Chapter 9

Conclusions

The incorporation of first-principles dynamic models in process operations provides an unprecedented potential to integrate the decision-making hierarchy. This integration can lead to more systematic and proactive operations and to significant improvements in profitability. However, the use of first-principles models in on-line operations also presents many computational challenges. The objective of this dissertation has been to propose computational strategies aiming at overcoming some of these challenges. In this chapter, we summarize our contributions and present suggestions for future work.

9.1 Thesis Summary and Contributions

A brief comparison of different approaches for DAE-constrained optimization presented in Chapter 2 demonstrates that the full-discretization approach presents a highly favorable computational complexity. This is mainly due to the fact that it avoids expensive and repetitive integrations of the DAE model and it allows to exploit the sparsity of the DAE model directly in the NLP formulation.

In Chapter 3 we have seen that with advances in interior-point NLP algorithms and sparse linear algebra solvers, the potential of the full discretization approach is likely to keep expanding in upcoming years. Nevertheless, as larger applications are considered, we have seen this approach can be hindered by long solution times and computer memory bottlenecks. The effect of the solution time is particularly relevant in on-line operations where time-critical solutions are needed. In addition, we have seen that the analysis of

solutions in specific applications such as parameter and state estimation and optimal control can become cumbersome when general NLP solvers are used. To overcome these limitations we have presented the following contributions:

- The proposal of a direct strategy to verify the satisfaction of sufficient second-order conditions (SSOC) through the inertia of the Karush-Kuhn-Tucker matrix arising in full-space interior-point solvers.
- The derivation of strategies to reuse the factorization of the KKT matrix at the solution to extract reduced Hessian information through inexpensive backsolves.
- The derivation of diverse NLP sensitivity strategies to obtain fast approximate solutions for large-scale parametric NLPs.

This basic set of computational tools has enabled advances in optimization-based tasks arising in the context of model-based operations. The main contributions are:

- **Parameter Estimation** - Chapter 4
 - The derivation of strategies to extract large-scale parameter covariance information from the KKT matrix and the proposal of formulating multi-set parameter estimation problems as generalized multi-scenario NLPs.
- **Nonlinear Model Predictive Control** - Chapter 5
 - The proposal, analysis and discussion of the advanced-step NMPC controller and the derivation of sufficient stability conditions.
- **Moving Horizon Estimation** - Chapter 6
 - The proposal, analysis and discussion of the advanced-step MHE estimator. The analysis of the optimality conditions of MHE problems to establish connections with recursive Kalman-like estimators and to derive strategies to extract state covariance estimation directly from the KKT matrix.

- **NMPC-Coupling** - Chapter 7

- Derivation of strategies for synchronization of advanced-step NMPC and MHE.

These computational strategies have been used to derive a model-based operations framework for industrial low-density-polyethylene (LDPE) reactors. Such a framework has been lacking due to the computational complexity of the associated LDPE tubular reactor models. In the context of this application, some of the accomplishments presented in Chapter 8 are:

- Proposal of an all-at-once solution strategy for parameter estimation problems. Refinement of parameters using industrial measurement data and systematic inference analysis.
- Design and implementation of NMPC controller and MHE estimator based on distributed LDPE reactor model.
- Computational analysis of scalability of full-discretization approach.

The publications resulting from this dissertation are [129, 127, 133, 128, 130, 131, 126, 132, 125].

9.2 Recommendations for Future Work

Challenging industrial applications like the LDPE tubular reactor process have motivated most of the developments of this dissertation. In addition, these applications have helped us to identify many potential areas of improvement. Some recommendations for future work are the following:

Interior-Point NLP Solvers

- We have seen that the factorization of the KKT matrix is the most expensive step in NLP solvers and limits the solution time. This is particularly relevant in on-line applications such as NMPC and MHE. Preliminary results on the use of different reordering strategies for direct linear solvers are encouraging. We believe that a more detailed analysis of these strategies is essential in order to solve larger problems where non-intuitive structures tend to appear. In particular, reordering strategies can be generalized to identify coarse structures that can be exploited *automatically* by the linear solver in shared-memory machines. The potential of this approach could be relevant and can be aided with advances in multi-core technology.
- In many of the NLPs arising in this dissertation we have seen that the inertia correction strategy can take several factorizations to compute the search step. This becomes expensive in large-scale problems. More efficient inertia correction strategies could be considered to make the solver faster. In particular, it could be helpful to stop the linear solver as soon as the wrong inertia is detected. Another potential strategy could be to use a trust-region framework to exploit directions of negative curvature more efficiently. Parameter estimation problems could be greatly benefited from this.
- It is still necessary to develop a more general post-optimal analysis framework for IPOPT. In particular, the framework should be targeted towards warm-starting and solving parametric NLPs. Such a framework could be useful in applications where IPOPT is used to solve NLP subproblems like in MINLP algorithms.

NMPC and MHE

- There exist multiple opportunities to expand the application scope of NMPC. Throughout this dissertation we have argued that the full-discretization approach provides fast solutions to large-scale problems. This is mainly due to the fact that the DAE model

does not need to be solved completely at each iteration. On the other hand, this advantage becomes a disadvantage from a robustness point of view. For instance, Hong and coworkers have noted that the simultaneous approach is more prone to numerical ill-conditioning and requires of globalization strategies more often to converge the problem [60]. This is due to the fact that the search step simultaneously optimizes and tries to converge the discretized DAE model which is, in many cases, a highly nonlinear set of equations. They have observed that the robustness of this approach can be significantly improved if the discretized DAE model is converged at each iteration (i.e. as in a feasible-path NLP solver). This is in fact one of the main reasons why single and multiple-shooting approaches tend to be more robust and also tend to take full Newton steps more often. On the other hand, this comes at the expense of increased computational times and potential infeasibility issues. Consequently, it is clear that there exists a trade-off between robustness and solution time. We believe that the exploration of *hybrid* DAE strategies could be helpful in solving larger problems not only faster but also more reliably.

- In previous work we have reported a strategy that allows to extend the solution of the background NMPC problems over multiple sampling times [131]. This can be done by reusing the KKT matrix to provide fast approximate feedback at multiple sampling times. These multi-step strategies allow the NLP solver to take more time to converge the problem. Consequently, they can handle very large models and/or strong disturbances. Although these strategies have shown to be efficient in practice, their stability properties have not been explored. In particular, there exists the open question of how much benefit can we obtain by using the multiple-step strategy as opposed to simply extend the sampling time. We believe that a rigorous stability analysis of this strategy can be helpful in obtaining more insight.
- Robust NMPC is an important area of research that can be benefited from the computational strategies proposed in this work. In particular, multi-scenario formulations

can handle large uncertainty spaces efficiently since they can be parallelized quite easily. However, stability results are lacking for this approach and need to be explored.

- The stability properties of the proposed advanced-step estimator have not been established so far and it is an important research topic. Recently, Alessandri and coworkers have derived stability conditions for MHE estimators in the face of bounded disturbances and approximation errors [3]. Since the advanced-step MHE strategy provides a natural framework to obtain rigorous performance bounds based on the uncertainty description, we believe that these stability results can be adapted to the specifics of the proposed estimator.

Model-Based Operations for LDPE Reactors

- The NMPC controller needs to be tested on more general dynamic scenarios. For instance, grade transitions are challenging in LDPE reactors because the entire temperature profile needs to be changed. In addition, the recycle system has not been modelled in this study and this has a strong impact on the dynamics of grade transitions.
- More general objective functions need to be considered in the controller. In particular, energy costs are relevant in LDPE processes. We also believe that the use of an economics-oriented controller can help to identify design bottlenecks and to guide retrofitting tasks.
- The fouling problem has been treated so far as an uncertain disturbance. However, the fouling rate can also be manipulated through appropriate control actions (e.g. thermal shocks, pressure shocks). In order to do this, predictive fouling models are required. With this, the NMPC controller can also be used as a long-term planner of fouling/defouling operations that can be interfaced with a lower-level controller.

- Communication of the NMPC controller with the scheduling layer can be established through multi-stage NMPC formulations. In particular, the scheduling layer provides the polymer quality set-point that the controller follows. In addition, the controller must simultaneously take fouling into account.

Bibliography

- [1] D. S. Achilias and C. Kiparissides. Towards the development of a general framework for modeling molecular weight and compositional changes in free radical copolymerization reactions. *J. M. S. -Rev. Macromol. Chem. Phys.*, C32:183–234, 1992.
- [2] J. Albuquerque, L. T. Biegler, and R. E. Kass. Inference in dynamic error-in-variable-measurement problems. *AIChE J.*, 43:986–996, 1997.
- [3] A. Alessandri, M. Baglietto, and G. Battistelli. Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44:1753–1765, 2008.
- [4] N. Arora and L. T. Biegler. Parameter estimation for a polymerization reactor model with a composite-step trust-region nlp algorithm. *Ind. Eng. Chem. Res.*, 43:3616–3631, 2004.
- [5] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, 1998.
- [6] G. Bader and U. Ascher. A new basis implementation for a mixed order boundary value ode solver. *SIAM J. Sci. Comput*, 8:483–500, 1987.
- [7] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, Cambridge, MA, 1974.
- [8] R. A. Bartlett. *New Object-Oriented Approaches to Large-Scale Nonlinear Programming for Process Systems Engineering*. Ph.D Thesis. Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA, 2001.

- [9] R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal. Quadratic programming algorithms for large-scale model predictive control. *Journal of Process Control*, 12:775–795, 2002.
- [10] R. D. Bartusiak. Nlmpc: A platform for optimal control of feed- or product-flexible manufacturing. In *Assessment and Future Directions of NMPC*, pages 367–282. Springer, Berlin, 2007.
- [11] S. Basu and Y. Bresler. The stability of nonlinear least squares problems and the cramer-rao bound. *IEEE Transactions on Signal Processing*, 48:3426–3436, 2000.
- [12] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.
- [13] J. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA, USA, 2001.
- [14] J. Betts and W. P. Huffman. Large-scale parameter estimation using sparse nonlinear programming methods. *SIAM J. Optim.*, 14:223–244, 2003.
- [15] L. T. Biegler. Efficient solution of dynamic optimization and nmpc problems. In *Nonlinear Model Predictive Control*, pages 219–244. Birkhäuser-Verlag, Basel, 2000.
- [16] L. T. Biegler, A. M. Cervantes, and A. Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.*, 57:575–593, 2002.
- [17] T. Binder, L. Blank, H.G. Bock, R. Burlisch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J.P. Schlöder, and O. von Stryk. Introduction to model based optimization of chemical processes on moving horizons. In *Online Optimization of large-scale Systems: State of the Art*, pages 295–339. Springer, 2001.
- [18] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *9th IFAC World Congress, Budapest*, 1984.

- [19] C. P. Bokis, S. Ramanathan, J. Franjione, A. Buchelli, M. L. Call, and A. L. Brown. Physical properties, reactor modeling, and polymerization kinetics in the low-density polyethylene tubular reactor process. *Ind. Eng. Chem. Res.*, 41:1017–1030, 2002.
- [20] A. Brandolin, P. Lacunza, L. Ugrin, and N. J. Capiati. High-pressure polymerization of ethylene and improved mathematical model for industrial tubular reactors. *Polym. React. Eng.*, 4:193–241, 1996.
- [21] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere, 1975.
- [22] A. Buchelli, M. L. Call, A. L. Brown, A. Bird, S. Hearn, and J. Hannon. Modeling fouling effects in ldpe tubular polymerization reactors. 1. fouling thickness determination. *Ind. Eng. Chem. Res.*, 44:1474–1479, 2005.
- [23] A. Buchelli, M. L. Call, A. L. Brown, A. Bird, S. Hearn, and J. Hannon. Modeling fouling effects in ldpe tubular polymerization reactors. 2. heat transfer, computational fluid dynamics, and phase equilibria. *Ind. Eng. Chem. Res.*, 44:1480–1492, 2005.
- [24] J. Busch, J. Oldenburg, M. Santos, A. Cruse, and W. Marquardt. Dynamic predictive scheduling of operational strategies for continuous processes using mixed-logic dynamic optimization. *Comp. Chem. Eng.*, 31:574–587, 2007.
- [25] C. Büskens and H. Maurer. Sensitivity analysis and real-time control of parametric control problems using nonlinear programming methods. In *On-line Optimization of Large-scale Systems*, pages 57–68. Springer-Verlag, Berlin, 2001.
- [26] C. Büskens and H. Maurer. Sensitivity analysis and real-time optimization of parametric nonlinear programming problems. In *On-line Optimization of Large-scale Systems*, pages 3–16. Springer-Verlag, Berlin, 2001.
- [27] A. M. Cervantes and L. T. Biegler. Optimization strategies for dynamic systems. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*. 2000.

- [28] W. Chen, D. J. Balance, and J. O'Reilly. Model predictive control of nonlinear systems: Computational burden and stability. *IEEE Proc. Control Theory Appl.*, 147:387–394, 2000.
- [29] J. E. Cuthrell and L. T. Biegler. Simultaneous optimization and solution methods for batch reactor control profiles. *Comput. Chem. Eng.*, 13:49–62, 1989.
- [30] P. F. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, 1984.
- [31] G. de Nicolao, L. Magni, and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In *Nonlinear Model Predictive Control*, pages 3–22. Birkhäuser-Verlag, Basel, 2000.
- [32] D. DeHaan and M. Guay. A new real-time perspective on non-linear model predictive control. *Journal of Process Control*, 16:615–624, 1996.
- [33] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996.
- [34] M. Diehl. Real-time optimization of large-scale nonlinear processes. *Ph.D Thesis, University of Heidelberg, Heidelberg, Germany*, 2001.
- [35] M. Diehl, H. G. Bock, and J. P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43, 5:1714–1736, 2005.
- [36] M. Diehl, R. Findeisen, H. G. Bock, J. P. Schlöder, and F. Allgöwer. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEEE Control Theory Appl.*, 152, 3:296–308, 2005.
- [37] I. S. Duff. Ma57 - a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30:118–144, 2004.

- [38] S. J. Wright E. A. Yildirim. Warm-start strategies in interior-point methods for linear programming. *SIAM J. Opt.*, 12:782–810, 2002.
- [39] T. F. Edgar. Control and operations: when does controllability equal profitability? *Comp. Chem. Eng.*, 29:41–49, 2004.
- [40] R. Faber, P. Li, and G. Wozny. Sequential parameter estimation for large-scale systems with multiple data sets: 1. computational framework. *Ind. Eng. Chem. Res.*, 42:5850–5860, 2003.
- [41] A. V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York, 1983.
- [42] R. Findeisen and F. Allgöwer. Computational delay in nonlinear model predictive control. *Proc. Int. Symp. Adv. Control of Chemical Processes ADCHEM 2003, Hong Kong*, 2004.
- [43] R. Findeisen, M. Diehl, T. Burner, F. Allgöwer, H. G. Bock, and J. P. Schlöder. Efficient output feedback nonlinear model predictive control. In *Proceedings of American Control Conference*, pages 4752–4757, 2002.
- [44] B. A. Finlayson. *Nonlinear Analysis in Chemical Engineering*. McGraw-Hill, New York, 1980.
- [45] A. Flores-Tlacuahuac, L. T. Biegler, and E. Saldivar-Guerra. Dynamic optimization of hips open-loop unstable polymerization reactors. *Ind. Eng. Chem. Res.*, 44:2659–2674, 2005.
- [46] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44:525–597, 2002.
- [47] R. Fourer, D. Gay, and B. Kernighan. *AMPL*. The Scientific Press, South San Francisco, 1993.

- [48] R. Franke and J. Doppelhamer. Integration of advanced model based control with industrial it. In *Assessment and Future Directions of NMPC*, pages 399–406. Springer, Berlin, 2007.
- [49] N. Ganesh and L.T. Biegler. A reduced hessian strategy for sensitivity analysis of optimal flowsheets. *AIChE J.*, 33:282–296, 1987.
- [50] J. Gondzio and A. Grothey. Reoptimization with the primal-dual interior point method. *SIAM J. Opt.*, 13:842–864, 2003.
- [51] S. Goto, K. Yamamoto, S. Furui, and M. Sugimoto. Computer model for commercial high-pressure polyethylene reactor based on elementary reaction rates obtained experimentally. *J. Appl. Pol. Sci.*, 36:21–40, 1981.
- [52] N. I. M. Gould and J. A. Scott. A numerical evaluation of hsl packages for the direct solution of large sparse, symmetric linear systems of equations. *ACM Trans. Math. Software*, 30:300–325, 2004.
- [53] A. Grancharova, T. A. Johansen, and P. Tondel. Computational aspects of approximate explicit nonlinear model predictive control. In *Assessment and Future Directions of NMPC*, pages 181–192. Springer, Berlin, 2007.
- [54] I. E. Grossmann, M. Erdirik-Dogan, and R. Karuppiah. Overview of planning and scheduling for enterprise-wide optimization of process industries. *Automatisierungstechnik*, 56:64–79, 2008.
- [55] I.E. Grossmann, S.A. van den Heever, and I. Harjunkski. Discrete optimization methods and their role in the integration of planning and scheduling. In *Proceedings of CPC-6, Tucson*, 2001.
- [56] M. Häfele, A. Kienle, M. Boll, and Schmidt C.U. Modeling and analysis of a plant for the production of low density polyethylene. *Comp. Chem. Eng.*, 31:51–65, 2006.

- [57] E. L. Haseltine and J. B. Rawlings. Critical evaluation of extended kalman filtering and moving horizon estimation. *Ind. Eng. Chem. Res.*, 44:2541–2460, 2005.
- [58] A. Helbig, O. Abel, and W. Marquardt. Structural concepts for optimization based control of transient processes. In *Nonlinear Model Predictive Control*, pages 295–312. Birkhäuser-Verlag, Basel, 2000.
- [59] G. A. Hicks and W. H. Ray. Approximation methods for optimal control synthesis. *Can. J. Chem. Eng.*, 40:522–529, 1971.
- [60] W. Hong, S. Wang, P. Li, G. Wozny, and L. T. Biegler. A quasi-sequential approach to large-scale dynamic optimization problems. *AIChE J.*, 52:255–268, 2006.
- [61] A. E. M. Huesman, O. H. Bosgra, and P. M. J. Van den Hof. Degrees of freedom analysis of economic dynamic optimal plantwide operation. In *Proceedings of 8th International Symposium on Dynamics and Control of Process Systems, Cancun, Mexico*, 2007.
- [62] G. T. Huntington. Advancement and analysis of a gauss pseudospectral transcription for optimal control problems. *Ph.D Thesis, Massachusetts Institute of Technology, Cambridge, USA*, 2007.
- [63] Z. P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869, 2001.
- [64] J. Kadam and W. Marquardt. Sensitivity-based solution updates in closed-loop dynamic optimization. In *Proceedings of 7th International Symposium on Dynamics and Control of Process Systems, Cambridge, USA*, 2004.
- [65] J. Kadam, W. Marquardt, M. Schlegel, T. Backx, O. Bosgra, and P.J. Brouwer. Towards integrated dynamic real-time optimization and control of industrial processes. In *Proceedings Foundations of Computer-Aided Process Operations, Cambridge, USA*, 2003.

- [66] S. Kameswaram and L. T. Biegler. Convergence rates for direct transcription of optimal control problems using collocation at radau points. *Computational Optimization and Applications*, 41:81–126.
- [67] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20:359–392, 1999.
- [68] S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for general class of constrained discrete-time systems: stability and moving-horizon approximations. *IEEE Trans. Auto. Cont.*, 57:265–293, 1988.
- [69] M. G. Kendall and Stuart. A. *The Advanced Theory of Statistics*. Griffin, London, 1973.
- [70] D. M. Kim and P. D. Iedema. Molecular weight distribution in low-density polyethylene polymerization: Impact of scission mechanisms in the case of a tubular reactor. *Chem. Eng. Sci.*, 59:2039–2052, 2004.
- [71] I. W. Kim, M. J. Liebman, and T. F. Edgar. A sequential error-in-variables method for nonlinear dynamic systems. *Comp. Chem. Eng.*, 15:663–670, 1991.
- [72] C. Kiparissides, A. Baltsas, S. Papadopoulos, J. P. Congalidis, J. R. Richards, M. B. Kelly, and Y. Ye. Free-radical ethylene copolymerization in high-pressure tubular reactors. *Ind. Eng. Chem. Res.*, 44:2592–2605, 2005.
- [73] C. Kiparissides, G. Verros, and J. MacGregor. Mathematical modeling, optimization and quality control of high-pressure ethylene polymerization reactors. *J. M. S. - Rev. Macromol. Chem. Phys.*, C33:437–527, 1993.
- [74] C. Kiparissides, G. Verros, A. Pertsinidis, and I. Goosens. On-line parameter estimation in a high-pressure low-density polyethylene tubular reactor. *AIChE J.*, 42:440–454, 1996.

- [75] H. Knuuttila, A. Lehtinen, and A. Nummila-Pakarinen. Advanced polyethylene technologies: Controlled material properties. *Adv. Polym. Sci.*, 169:13–27, 2004.
- [76] M. Lacunza, P. E. Ugrin, Brandolin A., and N. J. Capiati. Heat transfer coefficient in a high pressure tubular reactor for ethylene polymerization. *Polym. Eng. Sci.*, 38:992–1013, 1998.
- [77] C. D. Laird. *Structured large-scale nonlinear optimization using interior-point methods: applications in water distribution systems, Ph.D. Thesis*. Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA, 2006.
- [78] C. D. Laird and L. T. Biegler. Large-scale nonlinear programming for multi-scenario optimization. In *Proceedings of the International Conference on High Performance Scientific Computing, Hanoi, Vietnam*, 2006.
- [79] B. I. Lee and M. G. Kesler. A generalized thermodynamic correlation based on a three-parameter corresponding states. *AIChE J.*, 21:510–527, 1975.
- [80] D. B. Leineweber. Efficient reduced sqp methods for the optimization of chemical processes described by large sparse dae models. *University of Heidelberg, Heidelberg, Germany*, 1999.
- [81] W. C. Li and L. T. Biegler. Process control strategies for constrained nonlinear systems. *Ind. Eng. Chem. Res.*, 27:1421–1433, 1988.
- [82] M. J. Liebman, T. F. Edgar, and L. S. Lasdon. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Comp. Chem. Eng.*, 16:963–986, 1992.
- [83] J. Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice*, 11:847–858, 2003.

- [84] G. Luft, H. Bitsch, and H. Seidl. Effectiveness of organic peroxide initiators in the high-pressure polymerization of ethylene. *J. Macromol. Sci. -Chem.*, A11:1089–1112, 1977.
- [85] Abramowitz M. and Stegun I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 1972.
- [86] L. Magni and R. Scattolini. Robustness and robust design of mpc for nonlinear systems. In *Assessment and Future Directions of NMPC*, pages 239–254. Springer, Berlin, 2007.
- [87] T.E. Marlin and A.N. Hrymak. Real-time operations optimization of continuous processes. In *Fifth International Conference on Chemical Process Control (CPC-5)*, 1996.
- [88] D. Q. Mayne. Nonlinear model predictive control: Challenges and opportunities. In *Nonlinear Model Predictive Control*, pages 23–44. Birkhäuser-Verlag, Basel, 2000.
- [89] P. A. P. Moran. Estimating structural and functional relationships. *J. Multivar. Anal.*, 1:192–199, 1971.
- [90] K. R. Muske, E. S. Meadows, and J. B. Rawlings. The stability of constrained receding horizon control with state estimation. In *Proceedings of American Control Conference*, pages 2837–2841, 1994.
- [91] Z. K. Nagy, R. Franke, B. Mahn, and F. Allgöwer. Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework. In *Assessment and Future Directions of NMPC*, pages 465–472. Springer, Berlin, 2007.
- [92] A. Nikandrov and C.L.E. Swartz. Sensitivity analysis of lp-mpc cascade control systems. *Journal of Process Control, In Press*, 2008.
- [93] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, NY, 1999.

- [94] R. H. Nyström, I. Harjunkoski, and A. Kroll. Production optimization for continuously operated processes with optimal operation and scheduling of multiple units. *Comp. Chem. Eng.*, 30:392–406, 2006.
- [95] T. Ohtsuka. A continuation/gmres method for fast computation of non-linear receding horizon control. *IEEE Trans. Aut. Contr.*, 44:648–654, 1999.
- [96] N. M. C. De Oliveira and L. T. Biegler. An extension of newton-type algorithms for nonlinear process control. *Automatica*, 31:281–286, 1995.
- [97] A. Pertsinidis, E. Papadopoulos, and C. Kiparissides. Computer aided design of polymer reactors. *Comp. Chem. Eng.*, 20:S449–S454, 1996.
- [98] H. J. Pesch. Real-time computation of feedback controls for constrained optimal control problems. part 2: A correction method based on multiple shooting. *Optimal Control Applications & Methods*, 10:147–171, 1989.
- [99] M. Y. B. Poku and L. T. Biegler. Nonlinear optimization with many degrees of freedom in process engineering. *Ind. Eng. Chem. Res.*, 43:483–500, 2004.
- [100] V. V. Pontryagin, Y. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers Inc. New York, NY, 1962.
- [101] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.
- [102] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Trans. Automat. Contr.*, 48:246–258, 2003.
- [103] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *JOTA*, 99:723–757, 1998.

- [104] J. B. Rawlings and B. R. Bakshi. Particle filtering and moving horizon estimation. *Comp. Chem. Eng.*, 30:1529–1541, 2006.
- [105] W. H. Ray. On the mathematical modeling of polymerization reactors. *Polymer Reviews*, C8:1–56, 1972.
- [106] G. W. Reddien. Collocation at gauss points as a discretization in optimal control. *SIAM J. Control Optim*, 17:298–306, 1979.
- [107] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon based approach for least-squares state estimation. *AIChE J.*, 42:2209–2224, 1996.
- [108] L. O. Santos, P. Afonso, J. Castro, N. Oliveira, and L. T. Biegler. On-line implementation of nonlinear mpc: An experimental case study. *Control Engineering Practice*, 9:847, 2001.
- [109] L. O. Santos and L. T. Biegler. A tool to analyze robust stability for model predictive controllers. *Journal of Process Control*, 9:233–246, 1999.
- [110] O. Schenk, A. Wächter, and M. Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *J. Comp. Opt. and App.*, 36:321–341, 2007.
- [111] H. Seidl and G. Luft. Peroxides as initiators for high-pressure polymerization. *J. Macromol. Sci. -Chem.*, A15:1–33, 1981.
- [112] A. Sirohi and K. Y. Choi. On-line parameter estimation in a continuous polymerization process. *Ind. Eng. Chem. Res.*, 35:1332–1343, 1996.
- [113] S. Skogestad. Self-optimizing control: the missing link between steady-state optimization and control. *Comp. Chem. Eng.*, 24:569–575, 2000.
- [114] T. T. Kraus, P. Kuehl, L. Wirsching, H. G. Bock, and M. Diehl. A moving horizon state estimation algorithm applied to the Tennessee Eastman benchmark process. In

- Proceedings of IEEE Conference on Multisensor Fusion and Integration*, volume 44, pages 2541–2460.
- [115] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grossmann. Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE J.*, 53:2301–2315, 2007.
- [116] I. B. Tjoa and L. T. Biegler. Sequential parameter estimation for simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Ind. Eng. Chem. Res.*, 30:376–385, 1991.
- [117] T. Tosukhowonga, J. Min Lee, J. H. Lee, and J. Lu. An introduction to a dynamic plant-wide optimization strategy for an integrated plant. *Comp. Chem. Eng.*, 29:199–208, 2004.
- [118] A. Toumi, M. Diehl, S. Engell, H. G. Bock, and J. P. Schlöder. Finite horizon optimizing control of advanced smb chromatographic processes. *16th IFAC World Congress, Prague*, 2005.
- [119] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. part i - algorithmic framework. *Ind. Eng. Chem. Res.*, 33:2115–2123, 1994.
- [120] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. part ii - problems with path constraints. *Ind. Eng. Chem. Res.*, 33:2123–2133, 1994.
- [121] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Program.*, 106:25–57, 2006.
- [122] D. C. White. Online optimization: What have we learned? *Hydrocarbon Processing*, 77:55–69, 1998.

- [123] R. E. Young, R. D. Bartusiak, and R. W. Fontaine. Evolution of an industrial non-linear model predictive controller. *In Proceedings of CPC-6, Tucson*, 2001.
- [124] R. C. M. Zabisky, W. M. Chan, P. E. Gloor, and A. E. Hamielec. A kinetic model for olefin polymerization in high-pressure tubular reactors. a review and update. *Polymer*, 33:2243–2262, 1992.
- [125] V. M. Zavala and L. T. Biegler. Large-scale parameter estimation in low-density polyethylene tubular reactors. *Ind. Eng. Chem. Res.*, 45:7867–7881, 2006.
- [126] V. M. Zavala and L. T. Biegler. The advanced step nmpe controller: Stability, optimality and robustness. *Automatica, To Appear*, 2008.
- [127] V. M. Zavala and L. T. Biegler. Large-Scale Nonlinear Programming Strategies for the Operation of Low-Density Polyethylene Tubular Reactors. In *Proceedings of ESCAPE 18, Lyon, France*, 2008.
- [128] V. M. Zavala and L. T. Biegler. Optimization-based strategies for the operation of low-density polyethylene tubular reactors: Moving horizon estimation. *Submitted for Publication*, 2008.
- [129] V. M. Zavala, C. D. Laird, and L. T. Biegler. Fast solvers and rigorous models: Can both be accommodated in nmpe? *In Proceedings of IFAC Workshop on NMPC for Fast Systems*, 2006.
- [130] V. M. Zavala, C. D. Laird, and L. T. Biegler. A Fast Computational Framework for Large-Scale Moving Horizon Estimation. In *Proceedings of 8th International Symposium on Dynamics and Control of Process Systems, Cancun, Mexico*, 2007.
- [131] V. M. Zavala, C. D. Laird, and L. T. Biegler. Fast implementations and rigorous models: Can both be accommodated in nmpe? *Int. J. Robust Nonlinear Control*, 18:800–815, 2008.

- [132] V. M. Zavala, C. D. Laird, and L. T. Biegler. Interior-point decomposition approaches for parallel solution of large-scale parameter estimation problems. *Chem. Eng. Sci., In Press*, 2008.
- [133] V. M. Zavala, C. D. Laird, and L. T. Biegler. A moving horizon estimation algorithm based on nonlinear programming sensitivity. *Journal of Process Control, In Press*, 2008.

Appendix A

IPOPT Primal-Dual System

In this Appendix, we provide some details on the implementation and solution of the primal-dual system in IPOPT [121].

Consider the nonlinear programming problem of the form,

$$\begin{aligned} \min \quad & f(x) \\ & g_L \leq g(x) \leq g_U \\ & x_L \leq x \leq x_U \end{aligned} \tag{A.1}$$

where $x \in \Re^n$ are the primal variables with lower and upper bounds $x_L \in \Re^n$, $x_U \in \Re^n$. The inequality constraints $g : \Re^n \rightarrow \Re^m$ are bounded by $g_L \in \Re^m$ and $g_U \in \Re^m$.

After this problem has been communicated to IPOPT, the solver makes an explicit distinction between the equality (defined with $g_L = g_U$) and inequality constraints to give,

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d_L \leq d(x) \leq d_U \\ & x_L \leq x \leq x_U \end{aligned} \tag{A.2}$$

The equality constraints are represented by $c : \Re^n \rightarrow \Re^{m_c}$ and $d : \Re^n \rightarrow \Re^{m_d}$ denotes the inequality constraints with bounds $d_L \in \Re^{m_d}$ and $d_U \in \Re^{m_d}$ and $m = m_c + m_d$. Having done this, the current implementation of IPOPT reformulates the general inequality constraints

by adding slack variables and their corresponding bounds,

$$\begin{aligned}
& \min && f(x) \\
& \text{s.t.} && c(x) = 0 \\
& && d(x) - s = 0 \\
& && x - x_L \geq 0, \quad x_U - x \geq 0 \\
& && s - d_L \geq 0, \quad d_U - s \geq 0
\end{aligned} \tag{A.3}$$

with $s \in \Re^{n_d}$. As required by IPOPT, if a variable bound does not exist, the user sets the corresponding value to a large number ($-\infty$ or ∞). Nevertheless, for efficiency reasons, the solver ensures that only the relevant specified bounds ($x_L, d_L > -\infty$ and $x_U, d_U < \infty$) are actually taken into account. This is done by reformulating the problem to,

$$\begin{aligned}
& \min && f(x) \\
& \text{s.t.} && c(x) = 0 \\
& && d(x) - s = 0 \\
& && (P_x^L)^T x - x_L \geq 0, \quad x_U - (P_x^U)^T x \geq 0 \\
& && (P_d^L)^T d(x) - d_L \geq 0, \quad d_U - (P_d^U)^T d(x) \geq 0
\end{aligned} \tag{A.4}$$

where $P_x^L \in \Re^{n \times n_{xL}}$, $P_x^U \in \Re^{n \times n_{xU}}$, $P_d^L \in \Re^{m_d \times n_{dL}}$ and $P_d^U \in \Re^{m_d \times n_{dU}}$ are projection or permutation matrices between variables x and the inequalities $d(x)$ and their corresponding bounds. Symbols n_{xL}, n_{xU}, n_{dL} and n_{dU} represent the number of valid bounds. Accordingly, notice that the dimensions of x_L, x_U, d_L and d_U are also reduced.

In order to derive the primal-dual system, we define the Lagrange function of the reformulated NLP (A.4) as,

$$\begin{aligned}
\mathcal{L} = & f(x) + y_c^T c(x) + y_d^T (d(x) - s) - z_L^T ((P_x^L)^T x - x_L) - z_U^T (x_U - (P_x^U)^T x) \\
& - \nu_L^T ((P_d^L)^T d(x) - d_L) - \nu_U^T (d_U - (P_d^U)^T d(x))
\end{aligned} \tag{A.5}$$

where $y_c \in \Re^{m_c}$ and $y_d \in \Re^{m_d}$ are the Lagrange multipliers for the equality and inequality constraints, respectively; $z_L \in \Re^{n_{xL}}$ and $z_U \in \Re^{n_{xU}}$ are multipliers for the lower and

upper bounds of the x variables; and $\nu_L \in \Re^{n_{dL}}$ and $\nu_U \in \Re^{n_{dU}}$ are the bound multipliers corresponding to the slack variables (multipliers of inequality constraints).

After eliminating the bounds by adding a logarithmic barrier term to the objective function, the primal-dual optimality conditions of problem (A.4) are given by:

$$\begin{aligned}
\nabla_x \mathcal{L} = \nabla_x f(x) + J_c(x)^T y_c + J_d(x)^T y_d - P_x^L z_L + P_x^U z_U &= 0 \\
\nabla_s \mathcal{L} = -y_d - P_d^L \nu_L + P_d^U \nu_U &= 0 \\
Sl_x^L Z_L e - \mu e &= 0 \\
Sl_x^U Z_U e - \mu e &= 0 \\
Sl_d^L V_L e - \mu e &= 0 \\
Sl_d^U V_U e - \mu e &= 0 \\
c(x) &= 0 \\
d(x) - s &= 0
\end{aligned} \tag{A.6}$$

where $J_c^T \in \Re^{n \times m_c}$ and $J_d^T \in \Re^{n \times m_d}$ are the Jacobian matrices of the equality and inequality constraints and the diagonal matrices,

$$\begin{aligned}
Z_L &= \text{diag}(z_L) \\
Sl_x^L &= \text{diag}((P_x^L)^T x - x_L) \\
Z_U &= \text{diag}(z_U) \\
Sl_x^U &= \text{diag}(x_U - (P_x^U)^T x) \\
V_L &= \text{diag}(\nu_L) \\
Sl_d^L &= \text{diag}((P_d^L)^T d(x) - d_L) \\
V_U &= \text{diag}(\nu_U) \\
Sl_d^U &= \text{diag}(d_U - (P_d^U)^T d(x)) .
\end{aligned} \tag{A.7}$$

The optimality conditions (A.6) can be viewed as a set of nonlinear equations parameterized in μ . For the solution of this system, we can derive a sequence of Newton steps

obtained from the linearization of the above expressions,

$$\begin{aligned}
H\Delta x + J_c^T \Delta y_c + J_d^T \Delta y_d - P_x^L \Delta z_L + P_x^U \Delta z_U &= -\nabla_x \mathcal{L} \\
-\Delta y_d - P_d^L \Delta \nu_L + P_d^U \Delta \nu_U &= -\nabla_s \mathcal{L} \\
Z_L(P_x^L)^T \Delta x + Sl_x^L \Delta z_L &= -(Sl_x^L Z_L e - \mu e) \\
-Z_U(P_x^U)^T \Delta x + Sl_x^U \Delta z_U &= -(Sl_x^U Z_U e - \mu e) \\
V_L(P_d^L)^T \Delta s + Sl_d^L \Delta \nu_L &= -(Sl_d^L V_L e - \mu e) \\
-V_U(P_d^U)^T \Delta s + Sl_d^U \Delta \nu_U &= -(Sl_d^U V_U e - \mu e) \\
J_c \Delta x &= -c(x) \\
J_d \Delta x - \Delta s &= -(d(x) - s)
\end{aligned} \tag{A.8}$$

where $H \in \Re^{n \times n}$ is the Hessian matrix. The system of linear equations (A.8) has the following structure,

$$\begin{bmatrix}
H & 0 & J_c^T & J_d^T & -P_x^L & P_x^U & 0 & 0 \\
0 & 0 & 0 & -I & 0 & 0 & -P_d^L & P_d^U \\
J_c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
J_d & -I & 0 & 0 & 0 & 0 & 0 & 0 \\
Z_L(P_x^L)^T & 0 & 0 & 0 & Sl_x^L & 0 & 0 & 0 \\
-Z_U(P_x^U)^T & 0 & 0 & 0 & 0 & Sl_x^U & 0 & 0 \\
0 & V_L(P_d^L)^T & 0 & 0 & 0 & 0 & Sl_d^L & 0 \\
0 & -V_U(P_d^U)^T & 0 & 0 & 0 & 0 & 0 & Sl_d^U
\end{bmatrix}
\begin{pmatrix}
\Delta x \\
\Delta s \\
\Delta y_c \\
\Delta y_d \\
\Delta z_L \\
\Delta z_U \\
\Delta \nu_L \\
\Delta \nu_U
\end{pmatrix}
= -
\begin{pmatrix}
\nabla_x \mathcal{L} \\
\nabla_s \mathcal{L} \\
c(x) \\
d(x) - s \\
Sl_x^L Z_L e - \mu e \\
Sl_x^U Z_U e - \mu e \\
Sl_d^L V_L e - \mu e \\
Sl_d^U V_U e - \mu e
\end{pmatrix} \tag{A.9}$$

we will refer to this set of linear equations as the primal-dual system. In the current implementation of IPOPT, the primal-dual system is decomposed by eliminating the bound multipliers leading to the augmented linear system,

$$\begin{bmatrix}
H + D_x & 0 & J_c^T & J_d^T \\
0 & D_s & 0 & -I \\
J_c & 0 & 0 & 0 \\
J_d & -I & 0 & 0
\end{bmatrix}
\begin{pmatrix}
\Delta x \\
\Delta s \\
\Delta y_c \\
\Delta y_d
\end{pmatrix}
= -
\begin{pmatrix}
\nabla_x \bar{\mathcal{L}} \\
\nabla_s \bar{\mathcal{L}} \\
c(x) \\
d(x) - s
\end{pmatrix} \tag{A.10}$$

where,

$$\begin{aligned}
\nabla_x \bar{\mathcal{L}} &= \nabla_x f(x) + J_c^T y_c + J_d^T y_d + P_x^U (Sl_x^U)^{-1} \mu e - P_x^L (Sl_x^L)^{-1} \mu e \\
\nabla_s \bar{\mathcal{L}} &= -y_d + P_d^U (Sl_d^U)^{-1} \mu e - P_d^L (Sl_d^L)^{-1} \mu e \\
D_x &= P_x^L (Sl_x^L)^{-1} Z_L (P_x^L)^T - P_x^U (Sl_x^U)^{-1} Z_U (P_x^U)^T \\
D_s &= P_d^L (Sl_d^L)^{-1} V_L (P_d^L)^T - P_d^U (Sl_d^U)^{-1} V_U (P_d^U)^T.
\end{aligned}$$

Once the augmented linear system is solved, we can obtain step directions for the bound multipliers from,

$$\begin{aligned}
\Delta z_L &= -z_L + (Sl_x^L)^{-1} (\mu e - Z_L (P_x^L)^T \Delta x) \\
\Delta z_U &= z_U + (Sl_x^U)^{-1} (\mu e - Z_U (P_x^U)^T \Delta x) \\
\Delta \nu_L &= -\nu_L + (Sl_d^L)^{-1} (\mu e - V_L (P_d^L)^T \Delta s) \\
\Delta \nu_U &= \nu_U + (Sl_d^U)^{-1} (\mu e - V_U (P_d^U)^T \Delta s)
\end{aligned} \tag{A.11}$$

Appendix B

Riccati Decomposition NMPC

In this Appendix, we describe a backward Riccati decomposition strategy for the KKT system associated with the NMPC problem (5.10). This strategy has been useful in establishing connections between the advanced-step NMPC controller presented in Chapter 5 and Riccati-like regulators.

To start the discussion, we consider the KKT system,

$$\begin{aligned} \Delta z_0 &= \Delta \eta \\ \left. \begin{aligned} Q_l \Delta z_l + W_l \Delta v_l - A_l^T \Delta \lambda_{l+1} + \Delta \lambda_l &= 0 \\ W_l^T \Delta z_l + R_l \Delta v_l - B_l^T \Delta \lambda_{l+1} &= 0 \\ \Delta z_{l+1} - A_l \Delta z_l - B_l \Delta v_l &= 0 \end{aligned} \right\} l = 0, \dots, N-1 \\ Q_N \Delta z_N + \Delta \lambda_N &= 0 \end{aligned} \tag{B.1}$$

where $Q_N = \nabla_{zz}^N \mathcal{L} = \nabla_{zz}^N F$, $Q_l = \nabla_{zz}^l \mathcal{L}$, $W_l = \nabla_{zv}^l \mathcal{L}$, $R_l = \nabla_{vv}^l \mathcal{L}$, $A_l = \nabla_{z_l} f_l$, $B_l = \nabla_{v_l} f_l$. Here, we want to analyze how does the perturbation $\Delta \eta$ propagates through the prediction horizon. For this, we follow a Ricatti solution approach of the sensitivity equations.

Starting at $l = N$ we define $\Pi_N = Q_N$ so that,

$$\Pi_N \Delta z_N + \Delta \lambda_N = 0 \tag{B.2}$$

we propagate backwards in time through the dynamic equations,

$$\Delta \lambda_N = -\Pi_N A_{N-1} \Delta z_{N-1} - \Pi_N B_{N-1} \Delta v_{N-1} \tag{B.3}$$

we use the primal equations of Δv_{N-1} to put the control in explicit form,

$$\begin{aligned}
R_{N-1}\Delta v_{N-1} &= -W_{N-1}^T\Delta z_{N-1} + B_{N-1}^T\Delta\lambda_N \\
R_{N-1}\Delta v_{N-1} &= -W_{N-1}^T\Delta z_{N-1} - B_{N-1}^T\Pi_N(A_{N-1}\Delta z_{N-1} + B_{N-1}\Delta v_{N-1}) \\
\Delta v_{N-1} &= -(R_{N-1} + B_{N-1}^T\Pi_N B_{N-1})^{-1}(B_{N-1}^T\Pi_N A_{N-1} + W_{N-1}^T)\Delta z_{N-1}.
\end{aligned} \tag{B.4}$$

To establish the recursion we use the primal equations of Δz_{N-1} ,

$$\begin{aligned}
Q_{N-1}\Delta z_{N-1} + W_{N-1}\Delta v_{N-1} - A_{N-1}^T\Delta\lambda_N + \Delta\lambda_{N-1} &= 0 \\
Q_{N-1}\Delta z_{N-1} + W_{N-1}\Delta v_{N-1} + A_{N-1}^T(\Pi_N A_{N-1}\Delta z_{N-1} + \Pi_N B_{N-1}\Delta v_{N-1}) + \Delta\lambda_{N-1} &= 0 \\
(Q_{N-1} + A_{N-1}^T\Pi_N A_{N-1})\Delta z_{N-1} + (W_{N-1} + A_{N-1}^T\Pi_N B_{N-1})\Delta v_{N-1} + \Delta\lambda_{N-1} &= 0
\end{aligned} \tag{B.5}$$

plugging (B.4) in (B.6) gives,

$$\Pi_{N-1}\Delta z_{N-1} + \Delta\lambda_{N-1} = 0 \tag{B.6}$$

with,

$$\begin{aligned}
\Pi_{N-1} &= Q_{N-1} + A_{N-1}^T\Pi_N A_{N-1} \\
&\quad - (A_{N-1}^T\Pi_N B_{N-1} + W_{N-1})(R_{N-1} + B_{N-1}^T\Pi_N B_{N-1})^{-1}(B_{N-1}^T\Pi_N A_{N-1} + W_{N-1}^T).
\end{aligned} \tag{B.7}$$

Propagating backwards the recursion stops at the initial conditions,

$$\begin{aligned}
\Pi_0\Delta z_0 + \Delta\lambda_0 &= 0 \\
\Delta z_0 &= \Delta\eta
\end{aligned} \tag{B.8}$$

so that $\Delta\lambda_0$ depends only on the problem data,

$$\Delta\lambda_0 = -\Pi_0\Delta\eta \tag{B.9}$$

Having $\Delta\lambda_0$ and Δz_0 we recover the rest of the variables through the forward recursion,

$$\begin{aligned}
\Delta z_0 &= \Delta\eta \\
\Delta\lambda_0 &= -\Pi_0\Delta z_0 \\
\Delta v_l &= -(R_l + B_l^T\Pi_{l+1}B_l)^{-1}(B_l^T\Pi_{l+1}A_l + W_l^T)\Delta z_l \\
\Delta z_{l+1} &= A_l\Delta z_l + B_l\Delta v_l \\
\Delta\lambda_{l+1} &= -\Pi_{l+1}\Delta z_{l+1} \\
l &= 0, \dots, N
\end{aligned} \tag{B.10}$$

with,

$$\begin{aligned}
\Pi_N &= Q_N \\
\Pi_{l-1} &= Q_{l-1} + A_{l-1}^T\Pi_l A_{l-1} \\
&\quad - (A_{l-1}^T\Pi_l B_{l-1} + W_{l-1})(R_{l-1} + B_{l-1}^T\Pi_l B_{l-1})^{-1}(B_{l-1}^T\Pi_l A_{l-1} + W_{l-1}^T) \\
l &= N, \dots, 0
\end{aligned} \tag{B.11}$$

Appendix C

CSTR Reactor Model

In this Appendix, we present the model equations of the nonlinear continuous-stirred tank reactor (CSTR) presented by Hicks and Ray [59]. The conceptual reactor is sketched in Figure C.1. The reactor dynamic behavior is described the differential equations,

$$\frac{dz^c}{dt} = \frac{z^c - 1}{\theta} + k_0 z^c \exp \left[\frac{-E_a}{z^t} \right] \quad (\text{C.1a})$$

$$\frac{dz^t}{dt} = \frac{z^t - z_f^t}{\theta} - k_0 z^c \exp \left[\frac{-E_a}{z^t} \right] + \alpha v (z^t - z_{cw}^t). \quad (\text{C.1b})$$

The system involves two states $z = [z^c, z^t]$ corresponding to dimensionless concentration and temperature, and one control v corresponding to the cooling water flow rate. The model parameters are $z_{cw}^t = 0.38$, $z_f^t = 0.395$, $E_a = 5$, $\alpha = 1.95 \times 10^4$, $\theta = 20$, and $k_0 = 300$. These correspond to the cooling water temperature, feed temperature, activation energy, heat transfer coefficient, reactor residence time, and the pre-exponential factor.

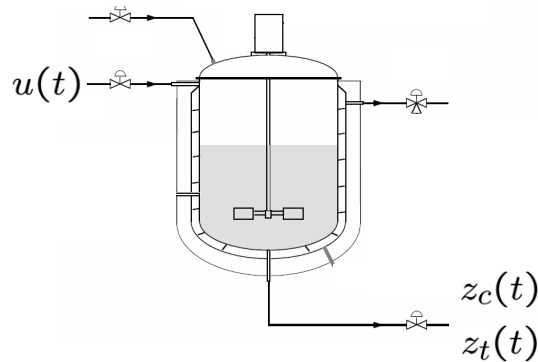


Figure C.1: Schematic representation of nonlinear CSTR reactor.

Appendix D

Riccati Decomposition MHE

In this Appendix, we describe a forward Riccati decomposition strategy for the KKT system associated with the MHE problem (7.3). This strategy has been useful in deriving strategies to extract MHE covariance information from general NLP solvers and to establish connections between the advanced-step MHE estimator presented in Chapter 6 and Kalman filters.

To start the discussion, we consider the KKT system,

$$\begin{aligned}
P_0 \Delta z_0 + W_0 \Delta w_0 - A_0^T \Delta \lambda_1 &= -r_{z_0} \\
P_l \Delta z_l + W_l \Delta w_l + \Delta \lambda_l - A_l^T \Delta \lambda_{l+1} &= -r_{z_l} \quad l = 1, \dots, N-1 \\
P_N \Delta z_N + \Delta \lambda_N &= -r_{z_N} \\
Q_l \Delta w_l + W_l^T \Delta z_l - G_l^T \Delta \lambda_{l+1} &= -r_{w_l} \quad l = 0, \dots, N-1 \\
\Delta z_{l+1} - A_l \Delta z_l - G_l \Delta w_l &= -r_{\lambda_{l+1}} \quad l = 0, \dots, N-1
\end{aligned} \tag{D.1}$$

Starting at time step $k = 0$,

$$\begin{aligned}
P_0 \Delta z_0 + W_0 \Delta w_0 - A_0^T \Delta \lambda_1 &= -r_{z_0} \\
Q_0 \Delta w_0 + W_0^T \Delta z_0 - G_0^T \Delta \lambda_1 &= -r_{w_0} \\
\Delta z_1 - A_0 \Delta z_0 - G_0 \Delta w_0 &= -r_{\lambda_1}
\end{aligned} \tag{D.2}$$

eliminating Δw_0 and Δz_0 ,

$$\begin{aligned}
\Delta w_0 &= -Q_0^{-1} W_0^T \Delta z_0 + Q_0^{-1} G_0^T \Delta \lambda_1 - Q_0^{-1} r_{w_0} \\
\Delta z_0 &= -(P_0 - W_0 Q_0^{-1} W_0^T)^{-1} (W_0 Q_0^{-1} G_0^T - W_0 A_0^T) \Delta \lambda_1 \\
&\quad + (P_0 - W_0 Q_0^{-1} W_0^T)^{-1} (W_0 Q_0^{-1} r_{w_0} - r_{z_0}) \\
\Delta z_0 &= -\Pi_0 (W_0 Q_0^{-1} G_0^T - W_0 A_0^T) \Delta \lambda_1 + \Pi_0 (W_0 Q_0^{-1} r_{w_0} - r_{z_0})
\end{aligned} \tag{D.3}$$

with $\mathbf{\Pi}_0 = (P_0 - W_0 Q_0^{-1} W_0^T)^{-1}$. Using the dynamic equations,

$$\begin{aligned}\Delta z_1 + (G_0 Q_0^{-1} W_0^T - A_0) \Delta z_0 - G_0 Q_0^{-1} G_0^T \Delta \lambda_1 &= -r_{\lambda_1} - G_0 Q_0^{-1} r_{w_0} \\ \Delta z_1 - \mathbf{M}_1 \Delta \lambda_1 &= -r_{\mathbf{M}_1}\end{aligned}\tag{D.4}$$

with,

$$\begin{aligned}\mathbf{M}_1 &= (G_0 Q_0^{-1} W_0^T - A_0) \mathbf{\Pi}_0 (W_0 Q_0^{-1} G_0^T - A_0^T) + G_0 Q_0^{-1} G_0^T \\ r_{\mathbf{M}_1} &= r_{\lambda_1} + G_0 Q_0^{-1} r_{w_0} - (G_0 Q_0^{-1} W_0^T - A_0) \mathbf{\Pi}_0 (r_{z_0} - W_0 Q_0^{-1} r_{w_0})\end{aligned}\tag{D.5}$$

Propagating to $k = 1$,

$$\begin{aligned}\Delta w_1 &= -Q_1^{-1} W_1^T \Delta z_1 + Q_1^{-1} G_1^T \Delta \lambda_2 - Q_1^{-1} r_{w_1} \\ \Delta z_1 &= -(P_1 - W_1 Q_1^{-1} W_1^T + \mathbf{M}_1^{-1})^{-1} (W_1 Q_1^{-1} G_1^T - A_1^T) \Delta \lambda_2 \\ &\quad + (P_1 - W_1 Q_1^{-1} W_1^T + \mathbf{M}_1^{-1})^{-1} (W_1 Q_1^{-1} r_{w_1} - r_{z_1} - \mathbf{M}_1^{-1} r_{\mathbf{M}_1}) \\ \Delta z_1 &= -\mathbf{\Pi}_1 (W_1 Q_1^{-1} G_1^T - A_1^T) \Delta \lambda_2 + \mathbf{\Pi}_1 (W_1 Q_1^{-1} r_{w_1} - r_{z_1} - \mathbf{M}_1^{-1} r_{\mathbf{M}_1})\end{aligned}\tag{D.6}$$

with $\mathbf{\Pi}_1 = (P_1 + \mathbf{M}_1^{-1} - W_1 Q_1^{-1} W_1^T)^{-1}$. So that,

$$\begin{aligned}\Delta z_2 + (G_1 Q_1^{-1} W_1^T - A_1) \Delta z_1 - G_1 Q_1^{-1} G_1^T \Delta \lambda_2 &= -r_{\lambda_2} - G_1 Q_1^{-1} r_{w_1} \\ \Delta z_2 - \mathbf{M}_2 \Delta \lambda_2 &= -r_{\mathbf{M}_2}\end{aligned}\tag{D.7}$$

with,

$$\begin{aligned}\mathbf{M}_2 &= (G_1 Q_1^{-1} W_1^T - A_1) \mathbf{\Pi}_1 (W_1 Q_1^{-1} G_1^T - A_1^T) + G_1 Q_1^{-1} G_1^T \\ r_{\mathbf{M}_2} &= r_{\lambda_2} + G_1 Q_1^{-1} r_{w_1} - (G_1 Q_1^{-1} W_1^T - A_1) \mathbf{\Pi}_1 (r_{z_1} + \mathbf{M}_1^{-1} r_{\mathbf{M}_1} - W_1 Q_1^{-1} r_{w_1})\end{aligned}\tag{D.8}$$

the process stops at $k = N$ where,

$$\begin{aligned}\Delta z_N - \mathbf{M}_N \Delta \lambda_N &= -r_{\mathbf{M}_N} \\ P_N \Delta z_N + \Delta \lambda_N &= -r_{z_N} \\ \Delta z_N &= -\mathbf{\Pi}_N (r_{z_N} + \mathbf{M}_N^{-1} r_{\mathbf{M}_N})\end{aligned}\tag{D.9}$$

This solution procedure can be generalized as,

$$\begin{aligned}
\Delta z_N &= -\mathbf{\Pi}_N(r_{z_N} + \mathbf{M}_N^{-1}r_{\mathbf{M}_N}) \\
\Delta \lambda_l &= \mathbf{M}_l^{-1}(\Delta z_l + r_{\mathbf{M}_l}) \\
\Delta z_{l-1} &= -\mathbf{\Pi}_{l-1}(W_{l-1}Q_{l-1}^{-1}G_{l-1}^T - A_{l-1}^T)\Delta \lambda_l + \mathbf{\Pi}_{l-1}(W_{l-1}Q_{l-1}^{-1}r_{w_{l-1}} - r_{z_{l-1}} - \mathbf{M}_{l-1}^{-1}r_{\mathbf{M}_{l-1}}) \\
\Delta w_{l-1} &= -Q_{l-1}^{-1}W_{l-1}^T\Delta z_{l-1} + Q_{l-1}^{-1}G_{l-1}^T\Delta \lambda_l - Q_{l-1}^{-1}r_{w_{l-1}} \\
l &= N, \dots, 1
\end{aligned} \tag{D.10}$$

where,

$$\begin{aligned}
\mathbf{\Pi}_0 &= (P_0 - W_0Q_0^{-1}W_0^T)^{-1} \\
\mathbf{M}_{l+1} &= (G_lQ_l^{-1}W_l^T - A_l)\mathbf{\Pi}_l(W_lQ_l^{-1}G_l^T - A_l^T) + G_lQ_l^{-1}G_l^T \\
\mathbf{\Pi}_{l+1} &= (P_{l+1} + \mathbf{M}_{l+1}^{-1} - W_{l+1}Q_{l+1}^{-1}W_{l+1}^T)^{-1} \\
\mathbf{\Pi}_N &= (P_N + \mathbf{M}_N^{-1})^{-1} \\
l &= 0, \dots, N-1
\end{aligned} \tag{D.11}$$

Appendix E

LDPE Tubular Reactor Model

In this appendix, we present a model that describes the dynamic behavior of the industrial LDPE tubular reactor described in Chapter 8. The model has been used to perform parameter estimation, state estimation and optimal control tasks.

E.1 Material Balances

The dynamics of the LDPE reactor core can be neglected since the time constant is small compared to that of the reactor wall and jacket [73]. With this, the design equations describing the evolution of the reacting mixture at the core of each zone become [74, 125],

Initiator(s)

$$\frac{\partial F_{I_i}}{\partial x} = -\frac{1}{\nu} k_{di} F_{I_i} \quad i = 1, \dots, N_I \quad (\text{E.1})$$

Monomer and Comonomer

$$\frac{\partial F_{m_i}}{\partial x} = -\frac{F_{m_i}}{\nu} \left(k_{Ii} C_R + \sum_{j=1}^2 k_{pji} \lambda_0^j + \sum_{j=1}^2 k_{f mji} \lambda_0^j \right) \quad i = 1, \dots, 2 \quad (\text{E.2})$$

Chain-transfer Agent(s)

$$\frac{\partial F_{S_i}}{\partial x} = -\frac{F_{S_i}}{\nu} \left(\sum_{j=1}^2 k_{sij} \lambda_0^j + \sum_{j=1}^2 k_{spij} \lambda_0^j \right) \quad i = 1, \dots, N_S \quad (\text{E.3})$$

Momentum Equation

$$\frac{\partial P}{\partial x} = -2f_r \frac{\rho \nu^2}{d_i} 1 \times 10^{-6} \quad (\text{E.4})$$

Long-Chain Branching (LCB), Short-Chain Branching (SCB)

$$\frac{1}{A} \frac{\partial F_{LCB}}{\partial x} = \sum_{i=1}^2 (k_{fpi1} + k_{fpi2}) \lambda_0^i \mu_1 \quad (\text{E.5})$$

$$\frac{\nu}{A} \frac{\partial F_{SCB}}{\partial x} = \nu \left(\sum_{i=1}^2 k_{bi} \lambda_0^i \right) + \frac{1}{A} \sum_{j=1}^{Ns} (k_{spj1} \lambda_0^1 + k_{spj2} \lambda_0^2) F_{sj} \quad (\text{E.6})$$

”Dead” Polymer Chains Moments

$$\frac{1}{A} \frac{\partial F_{\mu_0}}{\partial x} = \sum_{i=1}^2 \Delta_i \lambda_0^i + \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^2 k_{tcij} \lambda_0^i \lambda_0^j - \mu_1 \sum_{i=1}^2 \sum_{j=1}^2 k_{fpij} \lambda_0^i \quad (\text{E.7})$$

$$\frac{1}{A} \frac{\partial F_{\mu_1}}{\partial x} = \sum_{i=1}^2 C_{mi} \left(\sum_{j=1}^2 k_{pji} \lambda_0^j \right) \quad (\text{E.8})$$

$$\frac{1}{A} \frac{\partial F_{\mu_2}}{\partial x} = 2 \sum_{i=1}^2 \sum_{j=1}^2 k_{pji} \lambda_1^j C_{mi} + \sum_{i=1}^2 \sum_{j=1}^2 k_{tcij} \lambda_1^i \lambda_1^j \quad (\text{E.9})$$

Primary Radicals

$$F_R = \frac{\nu A}{(k_{I1} F_{m1} + k_{I2} F_{m2})} \left(\sum_{i=1}^{N_I} 2 \eta_i k_{di} F_{Ii} \right) \quad (\text{E.10})$$

”Live” Polymer Chains Moments

$$F_{\lambda_0^1} = \sqrt{\frac{(k_{I1} F_{m1} + k_{I2} F_{m2}) F_R}{(k_{td11} + k_{tc11}) + 2a(k_{td12} + k_{tc12}) + a^2(k_{td22} + k_{tc22})}} \quad (\text{E.11})$$

$$F_{\lambda_0^2} = a F_{\lambda_0^1} \quad (\text{E.12})$$

$$F_{\lambda_1^1} = \nu A \frac{\mathbb{B}_1 \Gamma_1^* - \mathbb{A}_2 \Gamma_1^*}{\mathbb{A}_1 - \mathbb{B}_1 \mathbb{B}_2} \quad (\text{E.13})$$

$$F_{\lambda_1^2} = \nu A \frac{-\Gamma_2^* - \mathbb{B}_2 \lambda_1^1}{\mathbb{A}_2} \quad (\text{E.14})$$

where,

$$a = \frac{(k_{fm12} + k_{p12}) F_{m2} + k_{fp12} F_{\mu 1}}{(k_{fm21} + k_{p21}) F_{m1} + k_{fp21} F_{\mu 1}} \quad (\text{E.15})$$

$$\mathbb{A}_1 = - \left(k_{p12} C_{m2} + \Delta_1 + \sum_{j=1}^2 k_{tc1j} \lambda_0^j \right) \quad (\text{E.16})$$

$$\mathbb{A}_2 = - \left(k_{p21} C_{m1} + \Delta_2 + \sum_{j=1}^2 k_{tc2j} \lambda_0^j \right) \quad (\text{E.17})$$

$$\mathbb{B}_1 = k_{p21}C_{m1} \quad (\text{E.18})$$

$$\mathbb{B}_2 = k_{p12}C_{m2} \quad (\text{E.19})$$

$$\Gamma_i^* = \left(k_{I_i}C_R + \sum_{j=1}^2 k_{fmji}\lambda_0^j + \sum_{j=1}^2 k_{pji}\lambda_0^j \right) C_{mi} \quad (\text{E.20})$$

$$+ \lambda_0^i \left(\sum_{j=1}^{Ns} k_{sji}C_{sj} + \sum_{j=1}^{Ns} k_{spji}C_{sj} \right) + \mu_2 \left(\sum_{j=1}^2 k_{fpji}\lambda_0^j \right) + (k_{\beta i} + k'_{\beta i})\lambda_0^i, \quad i = 1, \dots, 2 \quad (\text{E.21})$$

$$\Delta_i = \sum_{j=1}^2 k_{fmij}C_{mj} + \sum_{j=1}^{Ns} k_{sji}C_{sj} + \sum_{j=1}^2 k_{tdij}\lambda_0^j + \sum_{j=1}^2 k_{fpj}\mu_1 + k_{\beta i} + k'_{\beta i}, \quad i = 1, \dots, 2 \quad (\text{E.22})$$

Note that all the variables depend on both the space x and time t dimensions. The molar flow rate F_j for every component j in the mixture can be expressed in terms of the fluid velocity, ν , and its molar concentration C_j as,

$$F_j = \nu A C_j \quad (\text{E.23})$$

where A is the reactor cross-sectional area at a given axial position. Having the molar flow rates, the fluid velocity ν is calculated from the total molar flow rate and the mixture density (e.g. concentrations) from the above expression. The kinetic rate constants k are a function of temperature and pressure and are obtained from Arrhenius-type expressions of the form in (8.5). The density of the gas-polymer mixture ρ is computed from the Lee-Kesler equation of state and from complex empirical correlations. From the schematic representation of the reactor in Figure 8.1, we can see that the initial conditions of the differential equations at each zone are determined from material balances at the feed points.

Macromolecular properties of the polymer can be obtained in terms of the leading moments of the univariate chain-length distributions. Accordingly, the polymer number- and

weight-average molecular weights and polydispersity are given by,

$$MW_n = MW_0 \frac{F_{\mu_1}}{F_{\mu_0}} \quad (\text{E.24})$$

$$MW_w = MW_0 \frac{F_{\mu_2}}{F_{\mu_1}} \quad (\text{E.25})$$

$$PDI = \frac{MW_w}{MW_n} \quad (\text{E.26})$$

where MW_0 is the average molecular weight of a building unit in the polymer chain. The number of short- and long-chain branches per 1000 atoms can be obtained from,

$$LCB = 500 \frac{F_{LCB}}{F_{\mu_1}} \quad (\text{E.27})$$

$$SCB = 500 \frac{F_{SCB}}{F_{\mu_1}} \quad (\text{E.28})$$

The polymer density ρ_{pol} is correlated to the number of short-chain branches per 1000 carbon atoms,

$$\rho_{pol} = c_0 + c_1 SCB \quad (\text{E.29})$$

The polymer melt index can be correlated to the rest of the macromolecular properties as,

$$\log_{10}(MI) = c_2 + c_3 \log_{10}(MW_w) + c_4 \log_{10}(PDI) + c_5 \log_{10}(LCB). \quad (\text{E.30})$$

where c_j , $j = 0, \dots, 5$ are correlation parameters.

E.2 Energy Balances

We make use of Figure E.1 to illustrate the different components of the dynamic energy balance equations. The reacting mixture flowing inside the reactor is assumed to be at steady-state at all times. Accordingly, the evolution of reactor core temperature can be described by the following ordinary differential equation,

$$\rho c_p \nu \frac{\partial T}{\partial x} = \frac{\pi d^{in} \mathbf{U}^{in}}{A} (T^W - T) - \Delta H_r \quad (\text{E.31a})$$

$$T(t, 0) = T^{inlet}(t) \quad (\text{E.31b})$$

The axial dimension along each one of the zones is denoted by $x \in [0, x^L]$ where x^L is the total length of a particular zone k . The time dimension is denoted by symbol t . The above system of equations is defined for all the reactor zones $k = 1, \dots, N_Z$. The reacting mixture temperature at a particular zone is denoted by T and the wall temperature is represented by T^W . The gas-polymer mixture heat capacity is represented by c_p and is computed from thermodynamic relations. The overall heat of polymerization generated by chain propagation is represented by term ΔH_r . The overall heat transfer coefficient at the core-wall interface is represented by U^{in} and d^{in} is the inner diameter. The time-varying inlet temperature at a particular zone $T^{inlet}(t)$ is computed through boundary conditions consisting of energy balances at the feed points.

The reacting mixture is cooled down through a jacket cooling system. Due to the extremely high operating pressures, a thick stainless steel wall exists at the interface. Since this material presents a small capacitance, the dynamic responses of the reactor wall tend to dominate (order of minutes) and need to be captured by a dynamic model. In addition, due to the large thickness, there exists a temperature gradient across the radial position of the wall. A rigorous way to account for this would be to incorporate a two-dimensional energy balance along the axial and radial positions. However, this approach would result in a three-dimensional PDE defined at each zone, which would increase significantly the complexity of the reactor model. In order to avoid this, we follow the approach of Häfele and coworkers [56]. Here, we assume that the entire radial profile can be lumped into an average radial wall temperature. Accordingly, the wall temperature profile along the axial and time dimensions can be described by a parabolic PDE of the form,

$$\rho^W c_p^W \frac{\partial T^W}{\partial t} - \kappa^W \frac{\partial^2 T^W}{\partial x^2} = \frac{\pi d^{in} U^{in}}{A^W} (T - T^W) - \frac{\pi d^{out} U^{out}}{A^W} (T^W - T^J) \quad (\text{E.32a})$$

$$T^W(0, x) = T^{W,0}(x) \quad (\text{E.32b})$$

$$-A^W \kappa^W \frac{\partial T^W}{\partial x}(t, 0) = \dot{Q}_k^{in} \quad (\text{E.32c})$$

$$-A^W \kappa^W \frac{\partial T^W}{\partial x}(t, x^L) = \dot{Q}_k^{out} \quad (\text{E.32d})$$

where T^J denotes the temperature of the cooling water flowing along the jacket and $T^{W,0}(x)$

are the initial conditions of the wall temperature profile. Symbols ρ^W , c_p^W and κ^W denote the density, heat capacity and thermal conductivity of the stainless steel wall, respectively. The overall heat transfer coefficient at the interface between the wall and the jacket is denoted by \mathbf{U}^{out} , A^W is the cross-sectional area of the wall and the outer diameter is d^{out} . The axial inlet and outlet heat conduction flows across the wall at each zone k are denoted by $\dot{\mathbf{Q}}_k^{in}$ and $\dot{\mathbf{Q}}_k^{out}$, respectively. It is important to note that there exists heat flow continuity at the zone joints. Therefore, $\dot{\mathbf{Q}}_k^{in} = \dot{\mathbf{Q}}_{k-1}^{out}$, $k = 2, \dots, N_Z - 1$. At the reactor extremes, the heat flows $\dot{\mathbf{Q}}_0^{in}$ and $\dot{\mathbf{Q}}_{N_Z}^{out}$ can be calculated through natural convection expressions or can be assumed to be zero for a sufficiently long pipe.

Each reactor zone is equipped with an independent jacket cooling system where cooling water flows countercurrently to the reacting mixture inside the pipe. We assume plug flow in the jacket and negligible heat losses to the environment (perfect insulation). Accordingly, the dynamic evolution of the cooling water temperature at each zone can be described by the following first-order PDE,

$$\rho^J c_p^J \left(\frac{\partial T^J}{\partial t} - \nu^J \frac{\partial T^J}{\partial x} \right) = \frac{\pi d^{out} \mathbf{U}^{out}}{A^J} (T^W - T^J) \quad (\text{E.33a})$$

$$T^J(t, x^L) = T^{J,inlet}(t) \quad (\text{E.33b})$$

$$T^J(0, x) = T^{J,0}(x) \quad (\text{E.33c})$$

where ρ^J , c_p^J and ν^J are the density, heat capacity and velocity of the cooling water. The cross-sectional area of the jacket annulus is denoted by A^J . The time-varying inlet temperature of the cooling water is denoted by $T^{J,inlet}(t)$ and the initial conditions for jacket temperature profile are given by $T^{J,0}(x)$.

E.3 Model Notation

A = cross-sectional area, m^2

C_j = molar concentration of j th component, kgmol/m^3

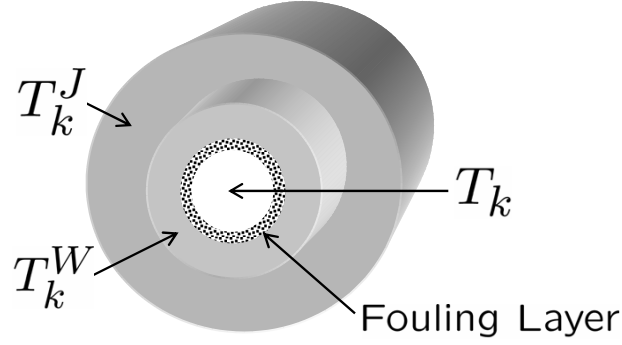


Figure E.1: Schematic representation of LDPE reactor core-wall-jacket interface.

c_p = reacting mixture heat capacity, kJ/kg·K

c_p^J = cooling agent heat capacity, kJ/kg·K

d = diameter, m

$D_{r,s}$ = concentration of "dead" polymer chains with r monomer units and s comonomer units, kmol/m³

F_j = molar flow rate of the j th component, kgmol/s

F_c = cooling agent flow rate, kg/s

f_r = Fanning friction factor

\mathbf{U} = heat transfer coefficient, kW/(m²K)

k_{bi} = intramolecular (backbiting) chain transfer of "live" polymer chains of type i , 1/s

k_{di} = decomposition rate constant of the i th initiator, 1/s

k_{Ii} = initiation rate constant for the i th monomer, m³/gmol·s

$k_{fmi j}$ = rate constant for the transfer of "live" polymer chains of type i to the j monomer
m³/gmol·s

k_{fpij} = rate constant for the transfer of "dead" polymer chains of type i to the j monomer
 $\text{m}^3/\text{gmol}\cdot\text{s}$

k_{pij} = propagation rate constant for the "live" polymer chains ending in the i th monomer unit with the j th monomer, $\text{m}^3/\text{gmol}\cdot\text{s}$

k_{sij} = rate constant for transfer of "live" polymer chains of type i to chain-transfer agent j , $\text{m}^3/\text{gmol}\cdot\text{s}$

k_{spij} = rate constant for incorporation of CTA j to "live" polymer chains of type i ,
 $\text{m}^3/\text{gmol}\cdot\text{s}$

k_{tcij} = termination by combination rate constant $\text{m}^3/\text{gmol}\cdot\text{s}$

k_{tdij} = termination by disproportionation rate constant $\text{m}^3/\text{gmol}\cdot\text{s}$

$k_{\beta i}$ = β -scission rate constant for secondary radicals, $1/\text{s}$

$k'_{\beta i}$ = β -scission rate constant for tertiary radicals, $1/\text{s}$

LCB = number of long-chain branches per 1000 carbon atoms

MW_n = number-average molecular weight, kg/kgmol

MW_w = weight-average molecular weight, kg/kgmol

MW_0 = molecular weight of a building unit, kg/kgmol

N_I = number of initiators

N_S = number of chain-transfer agent(s)

\mathbf{Q} = heat flow kJ/s

P = reactor pressure, atm

R_i = total concentration of "live" polymer chains ending in a radical of type i , kgmol/m^3

SCB = number of short-chain branches per 1000 carbon atoms

T = reactor temperature K

T^J = cooling agent temperature, K

T^W = wall temperature, K

ν = fluid velocity, m/s

Greek Letters

$-\Delta H_r$ = total heat of reaction, kJ/kgmol

η_i = efficiency of the i th initiator in the mixture

λ_n^i = n -th order single moment of the "live" polymer chains of type i , kgmol/m³

μ_n = n -th order single moment of the "dead" polymer chains, kgmol/m³

ρ_m = reacting mixture density, kg/m³

ρ_{pol} = polymer density, g/cm³

κ = thermal conductivity, kJ/m·K

Subscripts and Superscripts

I_i = i -th initiator

S_i = i -th chain-transfer agent

m_i = i -th monomer

R = primary radicals

W = wall

J = jacket

in = core

out = jacket